TRAVEL TIME FORECASTING USING PROBE VEHICLE DATA ON FREEWAYS

MASTER THESIS

Ricard Gardella Garcia

Facultat d'Informàtica de Barcelona
Universitat Politècnica de Catalunya

Advisor Mari Paz Linares Herreros - inLab FIB

Co-advisor Jamie Arjona Martínez - inLab FIB

Tutor Josep Casanovas Garcia

Tutor department Department of Statistics and Operations Research

Master Master in Innovation and Research in Informatics

Specialty Data Science

5th July 2019

Abstract

Nowadays, the percentage of connected cars is on the rise. Traffic forecasting plays a key role when optimizing mobility on freeways. Forecasting travel time is crucial for the improvement of traffic management systems, like route planners, which could mitigate the environmental and healthy problems related with traffic congestion. In addition, the increasing penetration of connected car makes Probe Vehicle Data a suitable data source to be exploited.

This master thesis deals with the problem of traffic forecasting on freeways using Probe Vehicle Data in the context of an educational cooperation agreement with the mobility research group of the inLab FIB and in the framework of a European Project, C-Roads.

The main goal of this master thesis is to perform travel forecasting using different models and evaluate these models with distinct conditions such as scenario, penetration of Probe Vehicle Data, prediction horizon, aggregation of the data and amount of data. Also, the state of the art shows that this problem could be solve using a wide spectrum of models like ARIMA, neural networks or linear regression.

The proposed experiments analyze the accuracy of the selected models and also examine how the different factors affect the forecasting. The data of the performed experiments have been generated by a microscopic traffic simulation tool, which simulates Probe Vehicle Data.

Finally, the obtained results show that some factors are highly dependent on the scenario and that the developed models with best results are ARIMA and neural networks for all tested scenarios.

Keywords: Traffic Forecasting, Traffic Simulation, Probe Vehicle Data, Time Series.

Acknowledgements

M'agradaria dedicar aquestes línies d'aquest treball de fi de màster a totes aquelles persones que, en més o menys mesura, m'han donat el seu suport i han aportat el seu gra de sorra en aquesta aventura. Considero però, que hi ha un conjunt de persones que sense el seu suport incondicional, aquest objectiu no s'hauria assolit de la mateixa manera.

En primer lloc, m'agradaria agrair a la Mari Paz la seva confiança i paciència infinites dipositades en mi. Des d'un bon principi ha estat al meu costat, incondicionalment, aportant el màxim valor, tant personalment com professionalment. Ningú podria demanar una millor directora de tesis. També a en Jamie, perquè és un absolut privilegi tenir un co-director al teu costat amb un coneixement tan aclaparador del qual sempre en pots aprendre alguna cosa nova. És una sort que dues persones així hagin dirigit la meva tesi, però sobretot, és una sort considerar aquestes dues persones com amics.

A en Josep, per l'oportunitat de pertànyer a l'inLab, per fer possible aquesta tesi i per l'ajuda brindada de cara a preparar la presentació. Em sembla un objectiu lloable el que es fa a l'inLab, ajudant a estudiants a preparar-se de cara al món laboral i que en Josep, com a director de l'inLab FIB, ho fa possible.

A tot l'equip de l'inLab, tant els estudiants actuals com els que ja han marxat, per convertir un lloc de treball un lloc del qual un mai marxaria. A la Marta per cuidar de tots. A Juan pels seus consells informàtics i passió desmesurada per la tecnologia que és un gust compartir. A en Víctor Sánchez pel seu humor i els nostres piques i memes.

Al meu company de feina, Albert, per les seves classes de matemàtiques, ajudar-me a entendre el que escric i ajudar-me a ser millor professional. És un plaer comptar amb algú que està en la teva mateixa situació i que fa el possible per ajudar-te, sempre crític, disposat a millorar. Ha sigut un plaer treballar amb tu.

A la meva família i amics. A la meva mare per estar sempre al meu costat de manera absolutament incondicional i donar-me suport en absolutament tot, sense tu això seria impossible. A la Sara, la meva segona mare, per estar sempre al meu costat passi el que passi. Al meu pare per preocupar-se sempre de com em va tot i que estic bé. I a la meva parella, Alba, per estimar-me tal com sóc, per treure'm sempre un somriure i per lluitar sempre el meu costat.

Per últim, m'agradaria agrair als que han marxat durant el transcurs d'aquest màster. Al meu avi Pere per ser un model a seguir, no només professional, sinó personal, i al meu gos Uruk per haver sigut el millor amic que un humà pot demanar. Us trobo molt a faltar.

Moltes gràcies a tots.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This Chapter introduces the context of this master thesis developed in educational cooperation agreement in the inLab FIB[1]. Also, the traffic forecasting on freeways using PVD problem is introduced. Furthermore, the main objectives and motivations of this master thesis are presented together with temporal planning of the whole thesis. Finally, this Chapter ends with an outline of the thesis.

1.1 Master Thesis Context

As the European Comision published in 2019[2], the development and large-scale deployment of Connected and Automated Mobility (CAM) provides a unique opportunity to make our mobility system safer, cleaner, more efficient and more user-friendly.

C-Roads[3], a project financially supported by the European Union, was created with the objective and motivation to upgrade the current infrastructure of the European freeways using new technologies. Inside the project of C-Roads, there is C-Roads Spain that includes C-Roads Catalan pilot, which is being deployed in the AP-7 freeway. Two objectives of the C-Roads Catalan pilot are to be able to forecast travel time and to detect and mitigate shock wave damping. InLab FIB is the research and innovation laboratory of the Barcelona School of Informatics at UPC, specialized in applications and services based on the latest ICT technologies is responsible for achieving these goals.

1.2 The Traffic Forecasting Using PVD Problem

The traffic forecasting problem is a very extensive problem. The most important thing when forecasting in traffic is the network that is being used. It is not the same as a network with only freeways than a network that also includes urban roads.

[1]https://inlab.fib.upc.edu/es

[2]https://ec.europa.eu/digital-single-market/en/connected-and-automated-mobility-europe

[3]https://www.C-Roads.eu/platform.html

We define forecasting or prediction as to the estimation of a response. Forecasting is the process of predicting future making use of the available data. For this project, forecasting will be done using different methods.

The problem that we are facing in C-Roads project is the case of freeways. Also, the C-Roads project is intended to work using Prove Vehicle Data (PVD) using sensors that will be installed in the freeway. Nowadays, those sensors are not installed yet and all data will be simulated using a traffic simulation tool, having in mind that the final objective is to use real data.

1.3 Objectives and Motivation

The main objective of this thesis is to perform travel forecasting in freeways using different methods with the data generated by a traffic simulation tool.

This problem could be solved using only simulation techniques, but those techniques are very expensive in contrast of the data-driven software solutions, so it is a motivation to explore new ways of solving this problem using data-driven solutions. With enough data, the data-driven solution can be cheaper than a simulation in terms of time, money and complexity.

Furthermore, another motivation exists to develop this thesis is that is developed for the inLab FIB smart mobility research group and also for a European Project.

In addition, the following research questions have been proposed:

- Which of the machine learning solutions proposed is better in terms of prediction and efficiency? There is a method that is always the best?
- Does the Neural Networks (NN) perform better than the simpler models like ARIMA or Linear Regression?
- How does the penetration of the PVD affects the traffic forecasting on freeways? Which is the minimal penetration needed in order to secure the predictions of a data-driven solution?
- How the prediction horizon affects the overall forecasting?
- What is the amount of data (in days) needed to perform accurate forecasting?
- Does the aggregation of the data affects the forecasting in a significant way?

1.4 Temporal Planning

According to the current academic regulations for the Master thesis of MIRI [4] this master thesis worth 90ECTS. The approximated workload for each ECTS is 30 hours so,

[4] https://www.fib.upc.edu/en/studies/masters/master-innovation-and-research-informatics/final-master-thesis

the workload of this thesis is 30 hours * 30 credits = 900 hours.

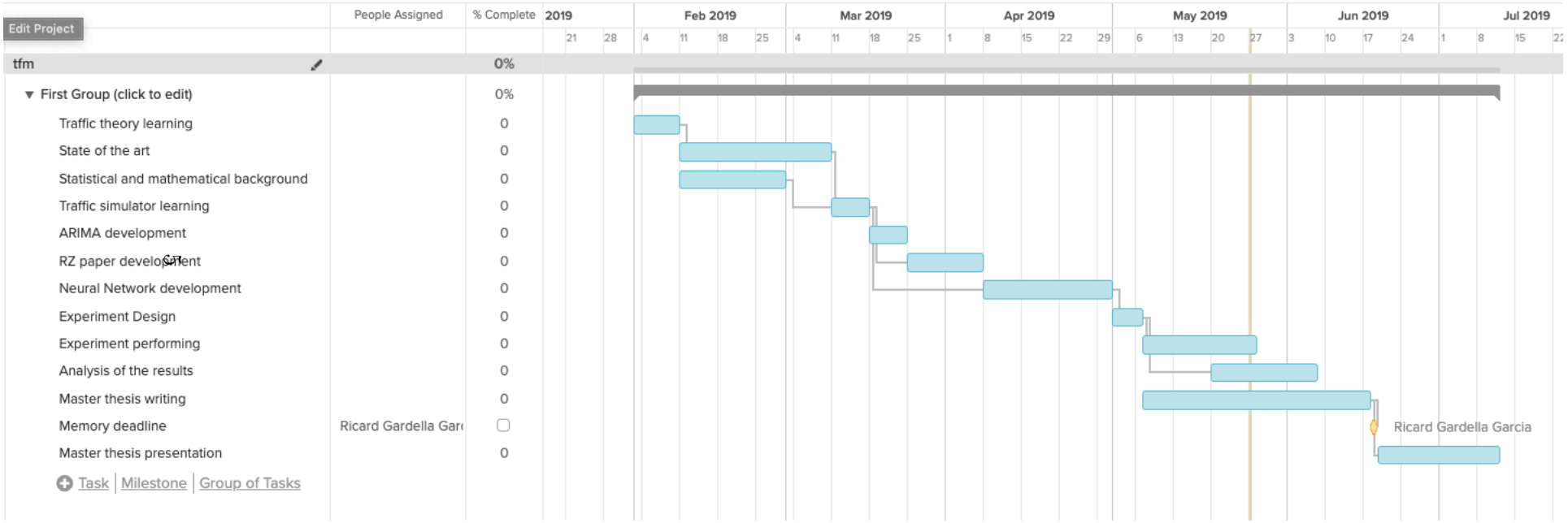
This is a type B master thesis and it has been developed under an education cooperation agreement with inLab FiB.

- Traffic theory learning: Before starting to write any state of the art, previous knowledge of traffic theory was needed.
- State of the Art: The first part of this master thesis is to review the existing literature and select the proposals that will be developed. This task also includes the writing of the state of the art that can be found in Chapter 2.
- Statistic and mathematical background: In order to understand the work done by other researchers in the traffic sector more mathematical and statistical knowledge was needed.
- Traffic simulator learning: This thesis uses a traffic simulation tool (Aimsun) to generate data. This task involves the usage learning of the tool.
- ARIMA development: ARIMA is the first method to be implemented in this thesis. This task also includes data preparation for this method.
- RZ paper development: RZ method is the second method to be implemented. This task also includes data preparation for this method.
- Neural network development: The last method to be implemented in this thesis is a Neural Network. This task also includes data preparation.
- Experiment design: This task includes all the design for comparison of all the method implemented.
- Experiment performing: Using the experiment design the experiments are performed.
- Analysis of the results: With the experiments performed, the results are analyzed and visualized.
- Master thesis writing: This task includes the writing of all the parts of the thesis that were not written before.
- Master thesis presentation: This task includes the presentation of the master thesis.

1.5 Master Thesis Outline

Considering the objectives exposed in this master thesis, the work done in this thesis is organized as follows:

- **Chapter 1** introduces the master thesis setting the objectives and planning of the master thesis.
- **Chapter 2** performs a review of the existing state of the art of traffic forecasting and makes a summary of the methods that are interesting to be implemented in this thesis.
- **Chapter 3** develops the selected algorithms from Chapter 2.
- **Chapter 4** presents the design of experiments of this thesis along with the computational experiments with the analysis of the results.
- **Chapter 5** presents the future work and conclusions of this thesis.


Edit Project
People Assigned
% Complete
2019
Feb 2019
Mar 2019
Apr 2019
May 2019
Jun 2019
Jul 2019
tfm
0%
First Group (click to edit)
0%
Traffic theory learning
0
State of the art
0
Statistical and mathematical background
0
Traffic simulator learning
0
ARIMA development
0
RZ paper development
0
Neural Network development
0
Experiment Design
0
Experiment performing
0
Analysis of the results
0
Master thesis writing
0
Memory deadline
Ricard Gardella Gar
Ricard Gardella Garcia
Master thesis presentation
0
Task
Milestone
Group of Tasks

Chapter 2

State of the Art

In this Chapter, the state of the art of the traffic forecasting with emphasis on freeways and PVD is discussed.

2.1 Introduction

Lana et al. [2018] presented an interesting summary about how the traffic forecasting methods are classified. Traffic forecasting methods can be classified in base of different aspects such as: prediction method, prediction horizon, prediction scale, prediction context, data source, exogenous factors, predicted variable, uni/multivariate, prediction performance, optimization type, computational effort, generalizability, scope of application, data resolution, and stream mining. Some of those characteristics are explained in detail below.

Prediction Method:

- Naive methods: The Naive methods are the simplest methods and do not make any model assumption. The typical Naive methods are the Instantaneous Travel Time, which supposes that current values remain constant indefinitely, and Historical Averages, which consider that the predicted value is the average of previous values under some filters such as day of the week or hour of the day. Those methods even that are very simple, are very useful.
- Parametric methods: The structure of the Parametric models is predetermined on the basis of theoretical considerations where the parameters are fitted using data. This kind of methods is based on traffic or theory and simulation on different levels. Traffic simulation models (macroscopic, mesoscopic, microscopic and three-phase traffic theory) and Time Series (ARIMA, linear regression, Kalman filtering, ATHENA, Maximum Likelihood and Prophet)
- Non-Parametric methods: The Non-Parametric methods refer to the models where the values of its parameters are determined from data. Examples of non-parametric

method are K-Nearest Neighbor, Neural Networks, Decision trees and SVM. Those methods tend to be the most expensive ones in terms of computing.

This taxonomy is explained in figure 2.1

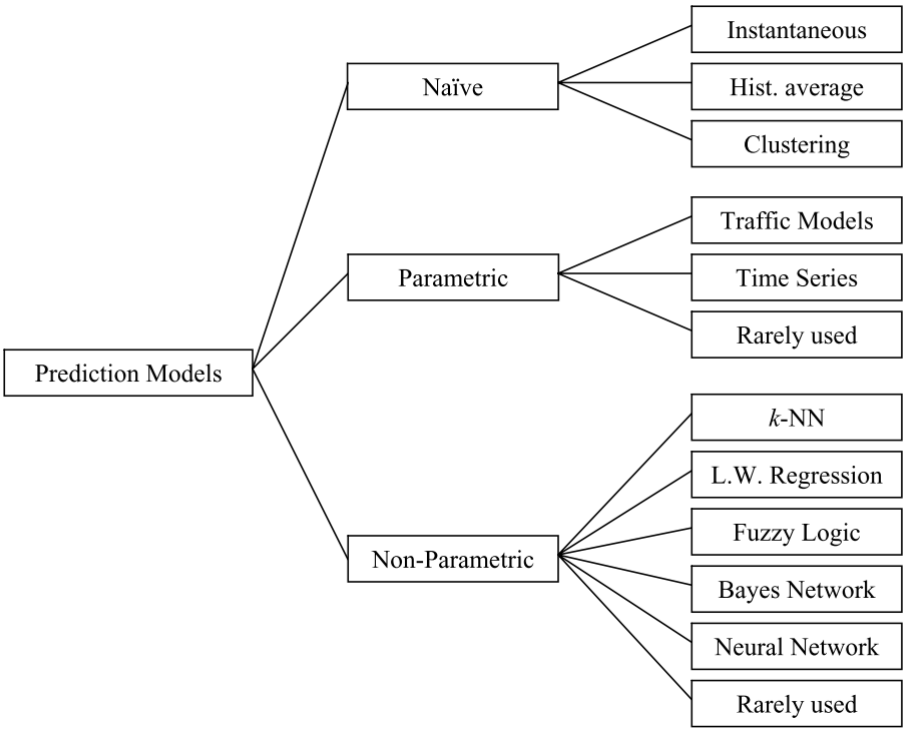


Figure 2.1: Taxonomy of prediction models. Source C.P.IJ. van Hinsbergen [2007]

Prediction Context:

- Urban: When the prediction context is urban we are usually talking about the travel time forecasting in cities with high traffic density.
- Rural.
- Freeway: Freeways is a very different context than the urban as must be taken into consideration.

Prediction Horizon:
Prediction horizon is how far in time you are predicting. Prediction horizon changes a lot depending on the prediction context that has been seen before.

- Long: In freeways, a long prediction horizon is in the range of more than 30 minutes.

- Short: In freeways, a short prediction horizon is in the range of one to 30 minutes.

Prediction Step:
The prediction step is the time that it takes between one prediction an another.

- Long: In freeways, a long prediction step is in the range of more than 10 minutes.
- Short: In freeways, a short prediction step is in the range of one to 10 minutes.

Data Source:
The data source can be another important factor to consider when we are doing forecasting in traffic. It can constrain the prediction method.

- Traffic management bureaus
- Automatic traffic recorders
- Sensors
- Cameras
- GPS-FCD (Floating Car Data)
- Cellphone data
- Public transport information
- Crowd sourcing
- Social media

2.2 Literature Review

In this section, the literature review has been done. This literature review contains papers with all the methods, contexts and data sources that were exposed in the section 2.1, but this state of the art does more emphasis on those papers that are in the context of a freeway and using GPS-PVD.

In terms of methods, according to the literature has been an evolution in recent years. The methods have evolved from naive and parametric to non-parametric methods. The most common method in the literature is a non-parametric method which is a method that solves time series. The Autoregressive Integrated Moving Average (ARIMA) is the time series model that is more used. There are a lot of different versions of ARIMA models, such ARIMA models that contemplate the seasonal aspects such as hour, day (seasonal ARIMA), ARIMA models that add spatiotemporal relations (Spatio-temporal ARIMA model) or ARIMA models that are developed by stock markets purposes. Nowadays in

the traffic forecasting literature, fewer studies propose these methods but they are still used as comparative methods.

The forecasting methods in traffic forecasting literature has evolved a lot. Thanks to the new sources of data and the enormous increase of computational power of the modern computers new algorithms that ten years ago were absolutely impossible to implement, nowadays a home-computer can, in terms of CPU power. The next paragraphs show how the trend has evolved from ARIMA models to machine learning models such as linear regression, neural networks, k-nearest neighbours and deep learning.

The data source of traffic forecasting literature is also evolving in recent years. Both in urban and freeway context the usage of the PVD (Probe Vehicle Data) giving the researches access to FCD (Floating Car Data) is increasing. The loop detectors are still being used in current projects but the future is clearly evolving to PVD and will be even more when the autonomous car arrives at the vast majority of the population.

In Treiber and Kesting [2013] we can see how the authors take a basic approach on the traffic time estimation but it is a more macroscopic approach rather than a microscopic approach. The authors show from the most simple formulas to the more advanced ones. They also provide different ways of getting the travel time estimation such as virtual trajectories, hybrid techniques, etc.

2.3 Parametric Methods

Rice and Van Zwet [2004], presents a method to predict the time that will be needed to traverse a given section of a freeway when the departure is at a given time in the future. The predictions are done on the basis of the current traffic situation in combination with historical data. They argue that the current traffic situation of a section of a freeway is well summarized by the current status travel time. This current status travel time can be estimated from single- or double-loop detectors, video data, probe vehicles, or any other means. This paper focuses on predicting or estimating the travel time with a given day and time and what they call time delay. They use different techniques in order to solve this problem and those techniques are linear regression, PCA analysis and nearest neighbours.

The research done by Ni and Wang [2008] show that is possible to construct a speed surface as a function of space and time. Then, one can reconstruct the trajectory of an imaginary vehicle by allowing it to adopt the local speed determined by the speed surface wherever the vehicle travels. This method proves useful if the trajectory method is used instead of the link method.

Jenelius and Koutsopoulos [2013], proposes a probabilistic principal components analysis model of network travel times. The model assumes that observed link travel times are projections from a lower- dimensional latent space, which determine the spatiotemporal correlations among links and time intervals. The paper gives us some algorithm and a real case applied in China. Even if the research is interurban can be applied in freeways.

Wan [2014], propose a citywide and real-time model for estimating the travel time

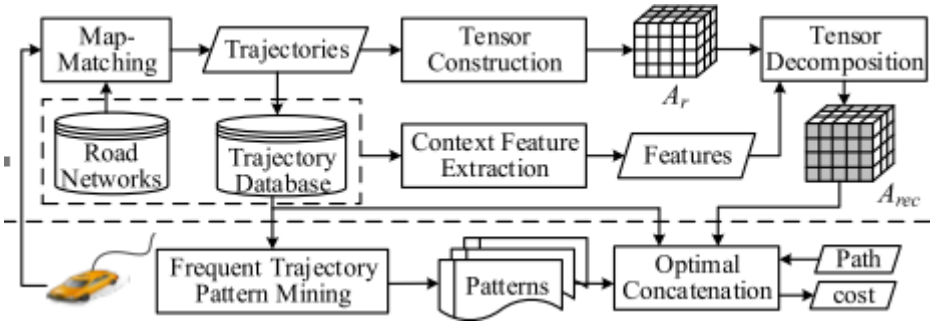


Figure 2.2: Framework of the paper. Source:Wan [2014]

of any path, in real time in a city, based on the GPS trajectories of vehicles received in current time slots and over a period of history as well as map data sources. The authors evaluated their method with extensive experiments generated by more than 32000 taxis for a period of two months.

The figure 2.2 shows the framework done by the authors in Wan [2014]

In Narayanan et al. [2015] examine the accuracy of popular travel time calculation techniques that use historical, instantaneous and predictive data to compute travel time. This paper tries to demonstrate that a dynamic approach is better than the previously implemented methods. The study is based on real-world traffic data collected in Singapore. They also analyze the possible variability of travel time that a user can have in its route during the trip. Their results show that a dynamic predictive routing using multiple prediction horizons provides a better estimate of the travel times as opposed to routing algorithms. The emphasis of this paper into a real problem is one of the strongest points of it.

In Jenelius and Koutsopoulos [2016], take a very practical approach to the travel time estimation using prove vehicle data. In this case, the authors use GPS data and its approach is in the interurban environment. The paper explains the model that is using a high level of detail of the links, sections, etc. The method the authors use in this research is Maximum Likelihood. The practical approach is useful and shows some results but it totally lacks any code or demonstration in how that results were obtained. In the end, the paper presented a statistical model for urban road network travel time estimation based on low-frequency GPS probe vehicle data. In this research, the component analysis (PPCA) model is proposed.

In Yang et al. [2016], how the authors realized that the congestion in developed and developing countries is an issue. In order to address that problem, they bed in intelligent systems to solve this problem. They address different computational approaches such as Neural Network (NN), generalized additive model (GAM) and autoregressive integrated moving average (ARIMA). The results of Yang et al. [2016] show how NN outperforms the other methods. The authors use a maximum horizon of an hour with a step of 15 minutes and have a data extension of 1 month. The data collected is FCD (Floating Car Data) used in a freeway scenario.

2.4 Non-Parametric Methods

In Ma et al. [2015], we can see how a different type of NN is used to solve the same problem but in a different context. In this case, the scenario is urban instead of a freeway and the data is collected using remote traffic microwave sensors. The authors use an LSTM with one input layer, one LSTM layer with memory blocks and one output layer because LSTM NN can automatically calculate optimal time lags. The authors of Ma et al. [2015] conclude that LSTM NN is able to learn time series with a long time dependency and outperforms GAM and ARIMA both in accuracy and stability. In order to validate those results, the authors used as a maximum horizon of 2 minutes with a data extension of 1 month with a step of 2 minutes. In Yi et al. [2017], presents a deep learning framework, which is a CNN that treats the problem using graph formulation. Experiments show that the model outperforms other methods such as LSTM and GRU on two real-world data-sets.

In Sun et al. [2018], instead of trying to improve the kNN method, it analyzes different kNN strategies in order to organize them and select the right one to improve prediction accuracy. The study is focused on select the parameter strategy, which is the way for the choice of kNN parameter values, and the data strategy, which is the way to separate training dataset. The authors suggest considering all parameter strategies simultaneously as ensemble strategies by tuning different parameters of kNN which are number os nearest neighbours, search step lengths and window size.

2.5 Summary and Conclusions

This chapter presents the state of the art in traffic forecasting. Most of the work exposed here is between 2004 and 2018 and most papers perform in freeways context and using GPD-FCD as the data source to perform the predictions. Other papers also have been considered.

Most common methods in the literature are time series (ARIMA), kNN and neural networks. Due to the simplicity of ARIMA and its easy implementation, ARIMA has been proposed as a solution in most of the old papers and as a comparative method in the new papers. It is clear though that the tendency changed a lot in the recent years, it can be seen that for example Rice and Van Zwet [2004] propose PCA, kNN and linear regression rather than use ARIMA besides being an old paper. This paper proposes a very interesting method using linear regression. This paper has been developed in the context of freeways as well, so it makes this research a very good candidate for implementation.

Other papers like Ni and Wang [2008] work using trajectories instead of links. Those researches have been discarded in the selection of algorithms to implement as trajectory method does not fit into the specifications of the C-ROADS project.

Maximum likelihood is proposed in Jenelius and Koutsopoulos [2016] and it is a very practical approach using PVD. Although this research is done in urban context instead of freeway stills very useful and could be a candidate for implementation.

In the recent researches, the rise of machine learning (kNN, Deep Learning, NN, LSTM) is very clear. In Ma et al. [2015] we can see how different types of NN are used to solve the same problem but in different contexts concluding that LSTM outperforms ARIMA.

Chapter 3

Selected Algorithms

This Chapter presents the selected algorithms that will be developed in this master thesis in order to solve the problem of travel time forecasting in freeways using PVD. The three algorithms that will be developed are ARIMA, a linear regression with varying parameters (RZ algorithm) and a Neural Network (NN).

Time series is a data with time dependency, where X_{t-1} precede X_t and X_{t+1} go after X_t. Most commonly, a time series is a sequence taken at successive equally spaced points in time. Examples of time series are heights of ocean tides, traffic demand in a freeway and the Bitcoin price evolution. Figure 3.1 shows the evolution of Bitcoin cryptocurrency in two and a half months, from May 9 until June 26. We can see how the evolution of the price of the Bitcoin is related to the time. Using the past data it could be possible to estimate or even try to forecast the future price of the Bitcoin.

3.1 ARIMA

As seen in Chapter 2, ARIMA is a method used to create models on time series and is one of the most used and well-known methods in the research literature for time series

Figure 3.1: Bitcoin evolution from May 9 until June 26. Source: Coinbase Pro

data, for that reasons has been implemented.

ARIMA stands for Autoregressive Integrated Moving Average, so from the name ARIMA three parts can be extracted.

- Autoregressive(AR): Autoregressive models and process are stochastic calculations in which future values are estimated based on a weighted sum of the past values
- Integrated(I): Represents the difference of raw observations to allow for the time series to become stationary.
- Moving Average(MA): A moving average for the residuals helps filtering the "noise" from random short-term fluctuations.

There are extensions and variations of the ARIMA. For example, SARIMA (Seasonal Auto-Regressive Integrated Moving Average) takes into account seasonal events, another variation is ARIMAX (Autoregressive Integrated Moving Average with Explanatory Variable) that combines time series with non-time series data.

The ARIMA has three parameters which must be calibrated in order to obtain accurate predictions:

- p: It is the number of autoregressive terms (AR). Allows the incorporation of past values into the ARIMA model.
- d: It is the number of non-seasonal differences.
- q: It is the number of coefficients forecast errors in the moving average part of the model (MA).Allow setting the error of the model as a linear combination of the error values observed at previous time points.

3.2 RZ Algorithm

This Section describes the algorithm that can be found on the paper Rice and van Zwet [2004] (RZ) and a small introduction about linear regression with varying parameters.

3.2.1 Introduction of the RZ algorithm

The objective of the work described in Rice and van Zwet [2004] is to predict travel time in freeways using spire data. This work makes the assumption that empirical observation has a linear relationship between future travel time forecasts and the current travel time.

The authors propose different methods for computing the travel time which are:

- Linear regression with varying parameters
- Principal Components

- Nearest Neighbors

The method implemented is the linear regression with varying parameters.

- D : set of days.
- L : set of loops.
- T : set of intervals of times in a day recorded in seconds.
- d_l : Distance from loop $l \in L$ to loop $l + 1 \in L$
- $v(d, l, t)$: Velocity measured on day $d \in D$ at loop $l \in L$ at time $t \in T$
- δ : Time lag which is the time that is added at your original starting time. The time lag is always greater or equal than zero.
- $T(d, t)$: Current travel time at day $d \in D$ at time $t \in T$.
- $T(d, t + \delta)$: Travel time at day $d \in D$ at time $t \in T$ with some time lag δ. This is the prediction variable.
- σ: It is a certain variance that the user needs to specify. The authors of the paper use a fixed value of ten minutes.

3.2.2 Introduction to Linear regression with Varying Parameters

In statistics, linear regression is a method to model the relationship between random variables as a linear one.

A linear regression is defined in the equation 3.1.

$$Y = \alpha + \beta X \tag{3.1}$$

where:

α is defining the X intercept, in other words, the value of Y when $X = 0$.

β as the slope of the line in the uni variate case.

X is the independent variables.

Y is the response.

Figure 3.2 shows an example of a linear regression.

When we execute a linear regression in any statistical package, the parameters α and β are optimized, but in this case, as we can see in section 3.2.3, the parameters α and β are allowed to vary in function of t and δ and we have to compute both α and β for each case. More information about linear models with varying parameters can be found in T.Hastie and Tibshirani [1994].

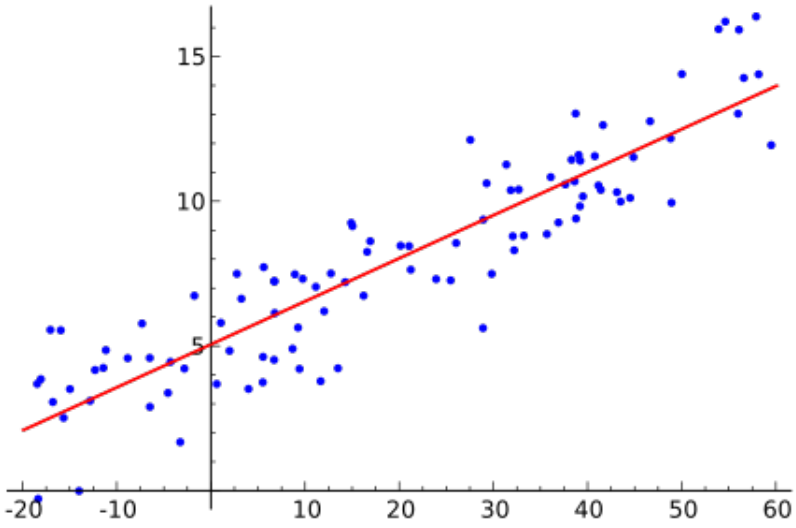


Figure 3.2: Linear regression. Source: Wikipedia.

3.2.3 Algorithm: Linear Regression with Varying Parameters

The goal is to predict $T(d, t+\delta)$ for $\delta >= 0$ where δ is the time lag. Considering $v(d, l, t)(d \in D, l \in L, t \in T)$ denoting the velocity that was measured on day d at loop l at time t. This array is our input data. With this data, we can compute two main equations. The current status travel time T^* is defined at equation 3.2 and is defined as T^* for some day d and time t. The authors consider all the loops $l \in L$ for computing the T^*, from $l = 1$ to $l = L - 1$.

$$T^*(d,t) = \sum_{l=1}^{L-1} \frac{2d_l}{v(d,l,t) + v(d,l+1,t)} \tag{3.2}$$

And the average historical travel time T is described at equation 3.3. We compute an average for the $T(d \in D, t)$ for a set of days D and for a certain $t \in T$.

$$\mathrm{T} = \frac{1}{\mid D \mid} \sum_{d \in D} T(d,t) \tag{3.3}$$

This paper assumes what is an empirical fact, that exists a relationship between the current status travel time $T^*(d,t)$ and the travel time in a future instant $T(d, t+\delta)$.

The model that the authors developed is the one that can be seen at equation 3.4 where α and β are parameters that have been extracted from the minimized function at equation 3.5. We need to set the parameters α and β for the linear regression because those parameters are allowed to vary in function of t and δ. In this model, ϵ is defined as some error. Those models are named Linear models with varying parameters [T.Hastie and Tibshirani [1994]]

$$T(d, t+\delta) = \alpha(t,\delta) + \beta(t,\delta)T^*(d,t) + \epsilon \tag{3.4}$$

The minimization function is described at equation 3.5. It is important to remark that in this equation, $s \in T$ means that s contains the time intervals contained in T. The notation changes to avoid confusion, because we need to introduce another parameter which will is $T(d,t)$, that measures the time travelled at some day, at some time.

$$\sum_{\substack{d \in D \\ s \in T}} (T(d,s) - \alpha(t,\delta) - \beta(t,\delta)T^*(d,t))^2 K(t+\delta-s) \tag{3.5}$$

Where K denotes the Gaussian density with mean 0 and a certain variance σ^2 which the user needs to specify and σ is the standard deviation of the data. K is defined in the equation 3.6. The value X in the equation 3.6 is the value product of the $t+\delta-s$ that can be seen at equation 3.5.

$$K(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-x^2}{2\sigma^2}} \tag{3.6}$$

Finally, using an optimization we obtain the values of the parameter α and β.

3.2.4 Adaptation of the RZ Algorithm

Previously, we have described the algorithm proposed by Rice and van Zwet [2004]. In our particular case, we made some modifications in the algorithm in order to solve our problem:

1. As we are using PVD, the concept of loop disappears. We will still use the notation $l \in L$ but a loop, in our context, is a whole section of the simulation model, not just a specific point in the road. We will have aggregated PVD data by link. More information about the data can be found in Section 4.2.

2. The authors predict a trajectory, which is a set of links. The predictions done in this master thesis using this algorithm are link predictions.

3. Another consequence of using sections and PVD instead of loops is that the $T^*(d,t)$ shown in equation 3.2 will change due that this formula is intended to work using loop detector data. Equation 3.7 shows the new formula for the $T^*(d,t)$ adapted for the PVD. In our case, as the forecast is done by link, S will only contain one link, but the equation has been formulated this way in order to perform trajectory forecast in future research. The $T^*(d,t)$ will be the following:

$$T^*(d,t) = \sum_{s=1}^{S} \frac{d_s}{v_s} \tag{3.7}$$

where:

s is the current section

S is the list that contains all the sections in the forecast

d_s is the distance of the current section at time t and day d

v_s is the speed of the current section at time t and day d

3.3 Neural Networks

An Artificial Neural Network (NN) is a machine learning method that takes inspiration about how the Biological Neural Network works. The NN can learn in the two variants of machine learning, which are supervised and non-supervised. The one used in this master thesis is supervised learning. The NN can learn to perform tasks by considering examples of the problem that the neural network is intended to solve.

A Neural Network can be represented as a directed graph, where the nodes are the neurons and the edges are the connections. Every neuron performs his own computation, getting an input vector as a parameter that is combined with the parameters (W and b), which is a variable that helps the model in a way that it can fit best for the given data, and outputs a scalar. The output can be the result of the model or give to another neuron. The computation of a neuron is the following:

$$output = f(X) = \sum_i (X_i W_i) + b \tag{3.8}$$

where:

$X \in \mathbb{R}^n$ are inputs

$W \in \mathbb{R}^n$ are coefficients.

$b \in \mathbb{R}$ is the bias.

3.3.1 Types of Neural Networks

The objective of this subsection is to cover the most important types of Neural Networks.

Perceptron

The perceptron is the simplest neuron model. The perceptron has n inputs, sum them all and applies an activation function. Then, pass the results to the output layer.

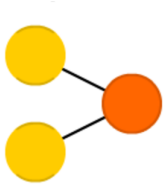


Figure 3.3: Perceptron architecture. Source: TowardsDataScience

Feed Forward Neural Network

This was the first type of NN. In this neural network architecture, the information only moves in one direction, from the input nodes through the hidden layer and then to the

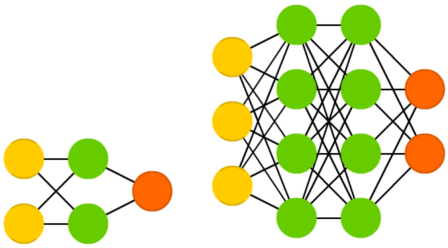


Figure 3.4: Feed Forward Neural Network (left) and Deep Feed Forward Neural Network (right). Source: TowardsDataScience

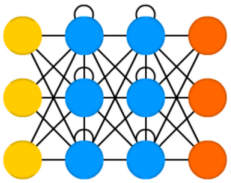


Figure 3.5: Recurrent NN architecture. Source: TowardsDataScience

output node. In this architecture, there are no cycles, loops or recurrence. A layer is a general term that applies to a collection of nodes operating together at a specific depth within a neural network.

The standard Feed Forward Neural Network only have one hidden layer, but the Deep Feed Forward Neural Networks or Multilayer Perceptron can have many more, this architecture of a neural network was the beginning of deep learning. Also, as all the neurons are interconnected, adding more layers lead to exponential training times but thanks to the optimizers this effect has been reduced a lot.

The layers that make a Feed Forward Neural network are the following:

- Input Layer: represents the data
- Hidden Layer: Layers between the input and the output where neurons take a set of weighted inputs and produce an output using an activation function.
- Output Layer: This last layer is the layer that gives the output.

Recurrent Neural Networks

This architecture introduce a different type of neurons which are the memory neurons, also, the graph is acyclic, letting loops in a neuron. Recurrent networks are distinguished from feed forward networks because those not only consider the present data also consider the recent past, and then combine to determine how they respond to new data. Figure 3.6 shows the structure of an unrolled recurrent neural network.

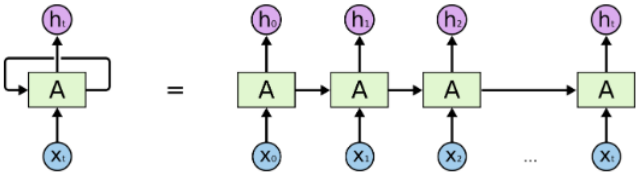


Figure 3.6: An unrolled recurrent neural network. Source: Colah

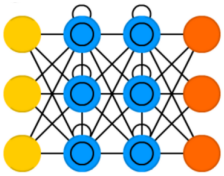


Figure 3.7: LSTM architecture. Source: TowardsDataScience

Long Short Term Memory

LSTM is a variation of the recurrent neural networks. The main problem of the Recurrent Neural Networks is that are unable to record the information of many frames ago. Sometimes the context needed is small and a standard RNN can solve the problem, but other times, more context is needed. For example, in the phrase "The roses are *red*", the context needed to predict the last word is small, but in the phrase "I like Apple products ... probably my new computer will be a *mac*" is possible that in this case, the gap between the relevant information and the point where is needed is really large. As this gap grows, Recurrent Neural Networks are unable to learn and connect the information. LSTM does not have this problem. Figure 3.7 shows the structure of a LSTM neural network.

LSTM improves RNN by the use of gate units that allow controlling the flow of the information helping with the vanishing gradient problem of the RNN. The vanishing gradient problem when a lot of layers using certain activation functions are added to neural networks, the gradients of the loss function approaches zero, making the network harder to train.

3.3.2 Activation Functions

The main piece of a neural network is the activation functions. The activation functions are responsible for transforming the input to output and also to secure that the output is non-linear. A NN without an activation function would be a linear regression. An activation function $f(x)$ is applied to the result of the combination seen at equation 3.8 and the result of this function is the output of the neuron. The most used activation function is ReLu, sigmoid and the Hyperbolic Tangent (tanh) are the classic activation functions.

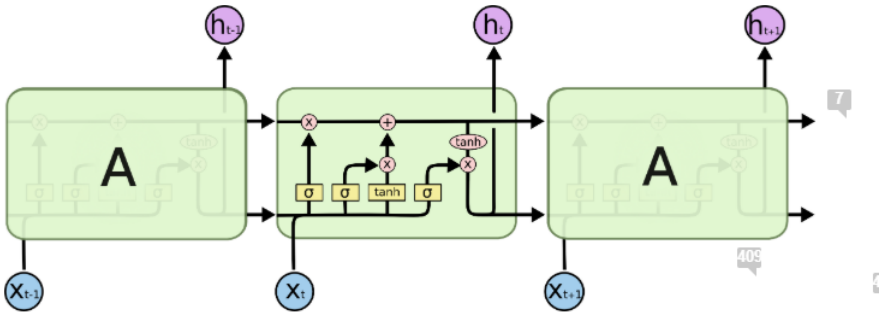


Figure 3.8: LSTM Module. Source: Colah

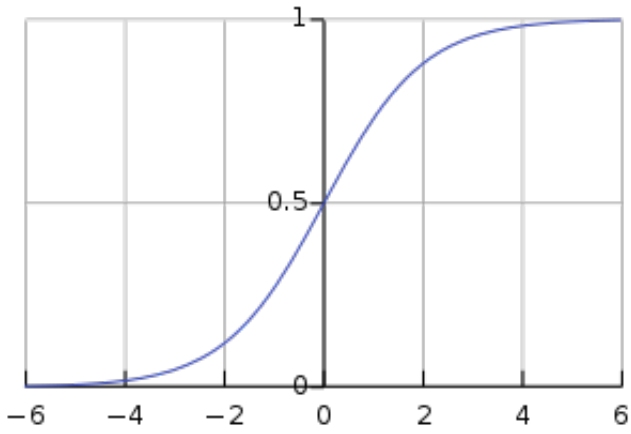


Figure 3.9: Sigmoid representation

The sigmoid is a mathematical function that follows an "S" curve. The sigmoid generates set of probabilities between 0 and 1 and it is widely used in binary classification problems. The equation 3.9 shows the mathematical representation of the sigmoid function and the figure 3.9 shows the graphical representation of the sigmoid function

$$Sigmoid(x) = \frac{e^x}{1 + e^x} \tag{3.9}$$

The other activation function that is popular is the tanh function. The tan-h is the alternative to the sigmoid function that was previously explained. The values of the tan-h go from -1 to 1. The equation 3.10 shows the equation for the tan-h and the figure 3.10 shows the shape of the activation function.

$$tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{3.10}$$

Although those activation functions, in this thesis another activation function will be used for the hidden layers, which is the Rectified Linear Unit (ReLu). As the sigmoid

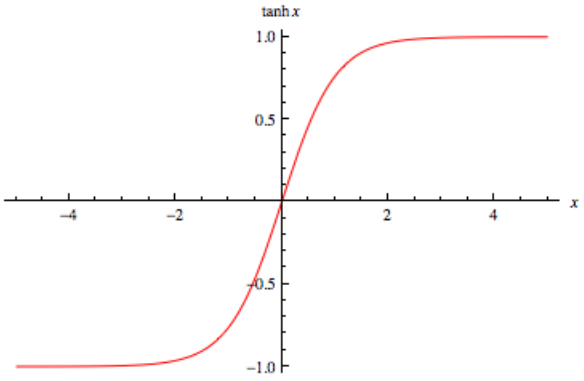


Figure 3.10: Hyperbolic Tangent representation

and tanh activation functions due to the vanishing gradient problem do not work well in networks with many layers. ReLu helps to solve the vanishing gradient problem, allowing models to learn faster and perform better. This is the function that is used for the new artificial neural networks in substitution of the sigmoid function. The function is described in equation 3.11 and the shape is shown at figure 3.11

$$ReLu(x) = max(0, x) \tag{3.11}$$

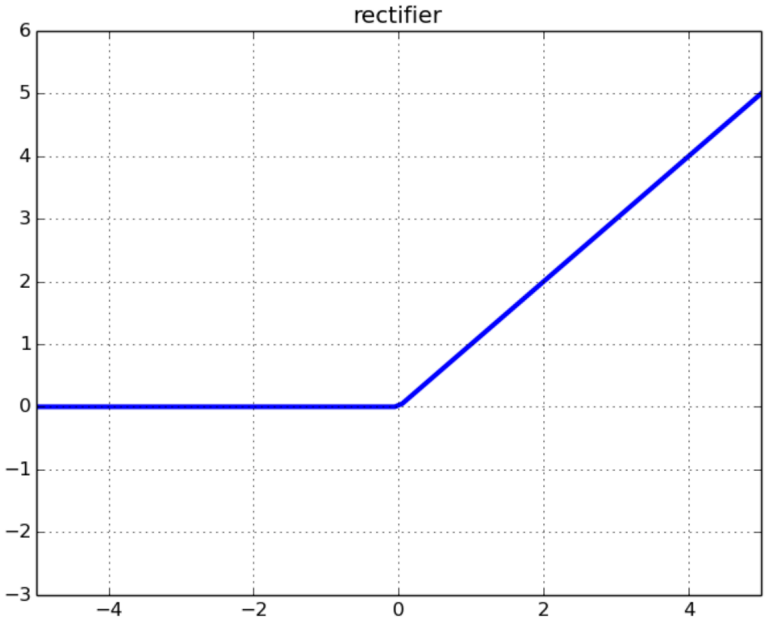


Figure 3.11: ReLu representation

There is no ”best” activation function, the choice of an activation function depends a

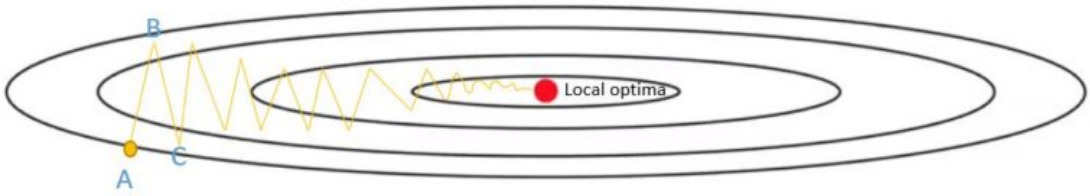


Figure 3.12: Stochastic Gradient Descent with momentum. Source: engMRK

lot in the problem we are trying to solve.

3.3.3 Optimizers

NN is a supervised machine learning algorithm this means that the model needs to be trained in order to perform predictions. This training process performs optimization in order to select the intern parameter values that minimize a loss function (error). One of the most important decision to make when creating a NN model is to decide which optimizer will be used and depends on the data.

Backpropagation

Most of the times, Neural Networks are trained using back propagation, that is a recursive and efficient method to compute the coefficient updates with the objective of improving the model. It uses gradient descend.

Stochastic Gradient Descent

Gradient descent is an optimization algorithm used to minimize some function by iterative moving in the direction of steepest descent as defined by the negative of the gradient. Stochastic means that the method uses randomly selected samples to evaluate the gradients. The size of every "step" is determined by the learning rate, while a higher learning rate can cover more each step, there is a risk of missing the lowest point. Stochastic Gradient descent maintains the same learning rate and does not change during the training period.

There is a variance of the Stochastic Gradient Descent which is the Stochastic Gradient Descent with Momentum that most of the times is faster and better than the standard Stochastic Gradient Descent. Stochastic Gradient Descent with momentum helps accelerate the gradients vectors in the right directions leading to faster convergence. figure 3.12 shows an example of a Stochastic Gradient Descent with momentum.

ADAM

ADAM stands for **ADA**ptive **M**oment estimation. ADAM is an optimization method that is a variant of the gradient descend. In contrary to the stochastic gradient descent

explained previously, ADAM does change the learning rate during the training period.

ADAM is a combination of the advantages of two other extensions of stochastic gradient descent which are:

- Adaptive Gradient Algorithm: Maintains a per-parameter learning rate that improves performance on problems with sparse gradients.
- Root Mean Square Propagation: Also maintains per-parameter learning rates that are adapted based on the average of recent magnitudes of the gradients for the weight.

Equation 3.12 shows the weights are computed in ADAM.

$$w_t = w_{t-1} - \eta \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon} \tag{3.12}$$

where:

w is model weights

η is the step size

$\hat{m}_t$ First moment

$\hat{v}_t$ Second moment

3.3.3.1 Neural Networks applied to time series.

The objective of this master thesis, as explained in Chapter 1, is to solve the problem travel time forecasting using PVD in freeways. The data that we have, explained in section 4.2 of this Chapter, are time series.

As seen in the literature in Chapter 2, one of the most widely used methods to solve the problem of this master thesis are neural networks. Different authors use different architectures of neural networks. In this master thesis, the neural network used is a multilayer feed forward neural network with backpropagation.

The main particularity that we have when doing neural networks with this type of data is that the sequence of values is very important. In this neural network, *look back* is introduced, which is an hyper parameter of the data when using a fixed-window approach. This parameter will take the previous time steps in order to predict the value of the next time period.

3.4 Implementation

This section will discuss all the implementation tools that have been used to develop the three algorithms explained in Chapter 3.

3.4.1 General Implementation Tools

Python[1] has been the programming language selected for the implementation because it is the programming language that is used in the whole C-Roads project and the language that the Aimsun Next API use. Although, other programming languages, such as Java, Scala or C# could have been used. The Python version that has been used is the latest, which is 3.7.

The data structures used in the implementation are instances of the library pandas[2], most of the times using the Pandas Dataframe[3] which are two-dimensional size-mutable, potentially heterogeneous tabular data structure with labelled axes and standard table data structure of two dimensions.

The IDE selected for the implementation is PyCharm[4] by Jetbrains[5], because it gives to the developer a very good environment for development.

In the implementation of these solutions, Git[6] have been used, specifically, GitLab[7] and Jenkins[8] have been used as a continuous integration tool.

3.4.2 ARIMA Implementation Tools

The implementation of the ARIMA itself is very straightforward. In order to execute the ARIMA model, a library has been used. In this case the *statsmodel*[9] library has been used. This library allows the parameter calibration of the p, d and q parameters and it is an open-source library released under the open source Modified BSD (3-clause) license.

3.4.3 RZ Algorithm Implementation Tools

The unique implementation singularity that these algorithms have is the minimization of the objective function described at equation 3.5. For that purpose, the library Scipy[10] has been used which is a collection of numerical algorithms and domain-specific toolboxes, including signal processing, optimization, statistics, etc.

[1] https://www.python.org
[2] https://pandas.pydata.org
[3] http://pandas.pydata.org/pandas-docs/version/0.19.1/generated/pandas.DataFrame.html
[4] https://www.jetbrains.com/pycharm/
[5] https://www.jetbrains.com
[6] https://git-scm.com
[7] https://about.gitlab.com
[8] https://jenkins.io
[9] http://www.statsmodels.org
[10] https://scipy.org/scipylib/

3.4.4 Neural Networks Implementation Tools

The Neural Network has been developed using TensorFlow [11] as a backend, which is an end-to-end platform that makes easy to build Machine Learning models. TensorFlow provides a direct path to production and does not constraint a programming language or device in order to use it. Also, running on top of TensorFlow, Keras[12] have been used, which is a high-level neural network API with main focus on fast experimentation. Keras runs on Python, supports both recurrent and convolutional neural networks or even a combination of recurrent and convolutional and also runs both on CPU and GPU.

[11] https://www.tensorflow.org
[12] https://keras.io

Chapter 4

Computational Experiments

This Chapter contains an introduction to the traffic simulation tool which has been used to generate the data, how the data has been aggregated, the performed computational experiments, the obtained results and a comparison between them.

4.1 Data Preparation

This Section gives an introduction to the traffic simulation tools, describes what network and demand have been used and how the data has been generated.

4.1.1 Introduction to the Traffic Simulation Tools

As said in Chapter 1, the objective of this master thesis is to forecast travel time on free roads using PVD. The data can be generated by a traffic simulation tool or by collecting the data from a real scenario using PVD from connected cars. Using a simulator to generate data, saves a lot of time an effort than collecting real data.

A traffic simulation tool is a software able to reproduce the traffic in order to develop meaningful operational strategies for real-time situations. For example, a traffic simulation tool could be used to simulate the impact of a roundabout in a city, how the increase of public transportation would affect the overall traffic or how more lanes in a freeway would affect the travel time between two points.

There are three main types of traffic simulation tools. Those are distinguished by the level of granularity.

- Microscopic: Microscopic simulators are simulators with the highest level of detail. The behaviour such as lane changing, car following, speed, etc., of every vehicle is simulated every time step and the information of every vehicle can be retrieved and modified.

- Macroscopic: Macroscopic simulators approach to focus on the complete road flow rather than in every vehicle, they are indented to simulate traffic flows for bigger

scenarios, the traffic is simulated as an aggregated flow. In this simulators, it is not possible to access to the data of every single vehicle or modify the behaviour of one concrete vehicle.

- Mesoscopic: Mesoscopic simulators are simulators that combine properties of both microscopic and macroscopic.

The data that is needed for this project is PVD, so for sure we need a microscopic simulator as PVD is data from each vehicle.

Given the project requirements, the previous expertise of the team in the software and the availability of the software license in the inLab, the simulator used is Aimsun (Advanced Integrated Microscopic Simulation and Urban Networks). Aimsun Next[1], which is a microscopic traffic simulation tool, has been used to generate the data necessary for the experiments. Aimsun SLU[2] also have mesoscopic and macroscopic simulators that will not be used in this project.

4.1.2 Traffic Network

The Catalan pilot of the C-Roads Spain project is done in the AP-7 so the traffic network used will be the AP-7 from La Roca del Vallès to La Junquera. Figure 4.1 can be seen the Aimsun model side by side with a standard Google Maps route from la Roca del Vallès to La Junquera. This Aimsun model was created by Aimsun SLU.

The main characteristics of the AP7 Aimsun model are:

- Size (km): 120.0
- Network type: Freeway
- Number of Centroids: 25
- Number of Detectors (i.e., toll locations): 50
- Number of sections: 532
- Number of nodes: 213
- Max Speed of the network: 33.33 m/s
- No signal controllers

In Figure 4.1, the centroid is the small circles that can be seen in every entrance-exit, En/Ex in the model. The centroids are responsible to generate and attract demand. It is important to remark that, in this model, in every centroid, there is a detector in the tolls

[1]https://www.aimsun.com/aimsun-next/

[2]https://www.aimsun.com

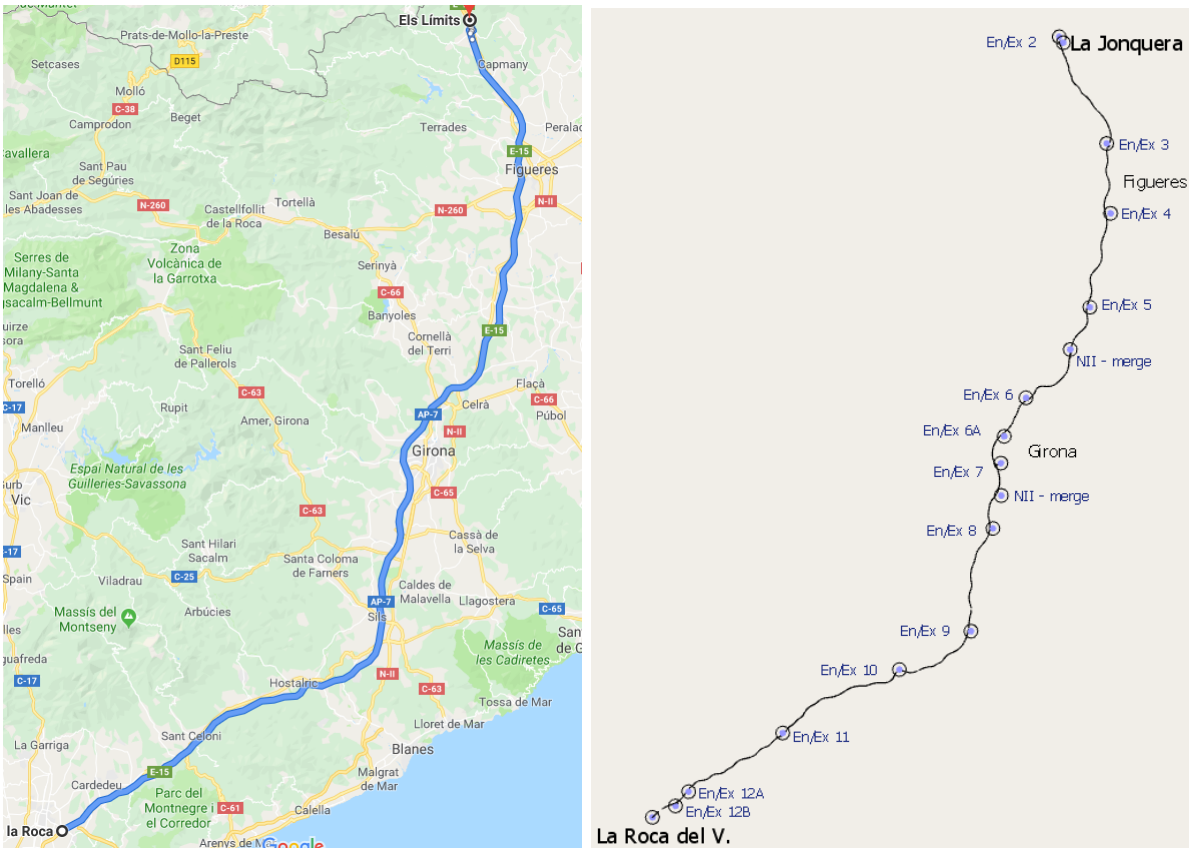


Figure 4.1: Google Maps (left) and Aimsun Model (right)

that collects the data. In figure 4.1 the tolls can be seen in the Aimsun model as a blue dot.

The major source of data used for the development of this traffic network are:

- Geometry infrastructure: Geographical representation of the network has been developed based on Open Street Map.
- Aerial data: Google Maps were used to verify information about roadway geometry.

4.1.3 Traffic Demand

The traffic demand is the number of travels that are performed in the whole simulation. The demand is divided into time intervals and is represented by a set of origin-destination matrices (OD matrices) that contains one matrix for each type of vehicle for each interval. Figure 4.2 shows an OD matrix for the vehicles of type Truck in Aimsun. The traffic demand of this Aimsun model has been created by Aimsun SLU with the data provided by Abertis Infraestructuras SA[3], collaborator of the C-Roads project and concessionaire of the tolls of the AP-7. The traffic demand duration is twenty-four hours and is composed of 96 OD matrices of 15 minutes each, for each type of vehicle. So it makes a total of 384 OD Matrices.

Four different demands have been created by Aimsun SLU, which are corresponding to different days of the week **Working Day, Friday, Saturday, Sunday**. The traffic demands follow an intuitive pattern, with a lot of traffic in hours like 8 am or 7 pm and with very low traffic during midnight, which is usually in a freeway during working days and Friday. In Saturday the lowest demand is experimented. Sunday pattern is also intuitive since a lot of people go back to Barcelona on Sunday. Figure 4.3 shows the number of vehicles that exits La Roca toll, plotting the four demands in the model. It can be seen that the demands **WorkDay** and **Friday** follow a similar pattern and the demands of **Saturday** and **Sunday** follow a different pattern.

The demand that this masters thesis will use in order to perform the experiments is the **WorkDay** demand, with 173.196 vehicles. The two types of vehicles which are divided in those who use Teletac[4] and those who not. Table 4.1 show the types of vehicles together with the number of vehicles for every 24h simulation.

Type of vehicle	**Number of vehicles**
Trucks	3.685
Trucks with Teletac	31.787
Cars	82.373
Cars with Teletac	55.350

Table 4.1: Vehicles for WorkDay

[3]https://www.abertis.com

[4]https://es.wikipedia.org/wiki/VIA-T

Matriz OD: 1268574, Nombre: 1. WorkDay C. Departure demand - Camion 05:00:00, ID Externo: C. Departure demand {468bb6c2-5236-4d... ? ×

Principal | Celdas | Histograma | Asignación de caminos | Parámetros

Cabeceras: ID: Nombre Categoría de Agrupación: Ninguno

☐ Permitir Valores Negativos ☐ Mostrar Todos los Centroides ☐ Ocultar Filas Vacías ☐ Ocultar Columnas Vacías

	1237635: 102	1237636: 103	1237687: 114	1237688: 115	1237703: 113	1237707: 112	1237710: 111
1237635: 102		0,52	0,02	0,01		0,01	
1237636: 103	0,47		0,39	0,73	0,13	0,11	0,02
1237687: 114	0,01	0,31		0,59	0,04	0,03	0,01
1237688: 115	0,04	0,55	0,60		0,24	0,06	0,02
1237703: 113		0,06	0,08	0,41		0,01	0,01
1237707: 112	0,01	0,10	0,13	0,12	0,03		0,01
1237710: 111		0,02		0,02	0,01	0,01	
1237713: 119		0,02					
1237716: 110	0,02	0,17	0,01	0,01		0,02	
1237719: 118	0,01	0,04	0,02	0,01		0,01	
1237722: 109	0,03	0,06	0,02	0,07	0,01	0,03	0,01
1237725: 117			0,08	0,13	0,04	0,04	0,01
1237728: 108		0,06	0,03	0,03		0,01	
1237731: 107	0,04	0,27		0,03	0,01	0,01	
1237735: 106	0,01	0,14		0,04			
1237738: 105	0,03	0,15					
1237742: 104		1,22	0,01	0,01			
Total	0,66	3,69	1,39	2,23	0,49	0,34	0,07

Figure 4.2: OD Matrix in Aimsun

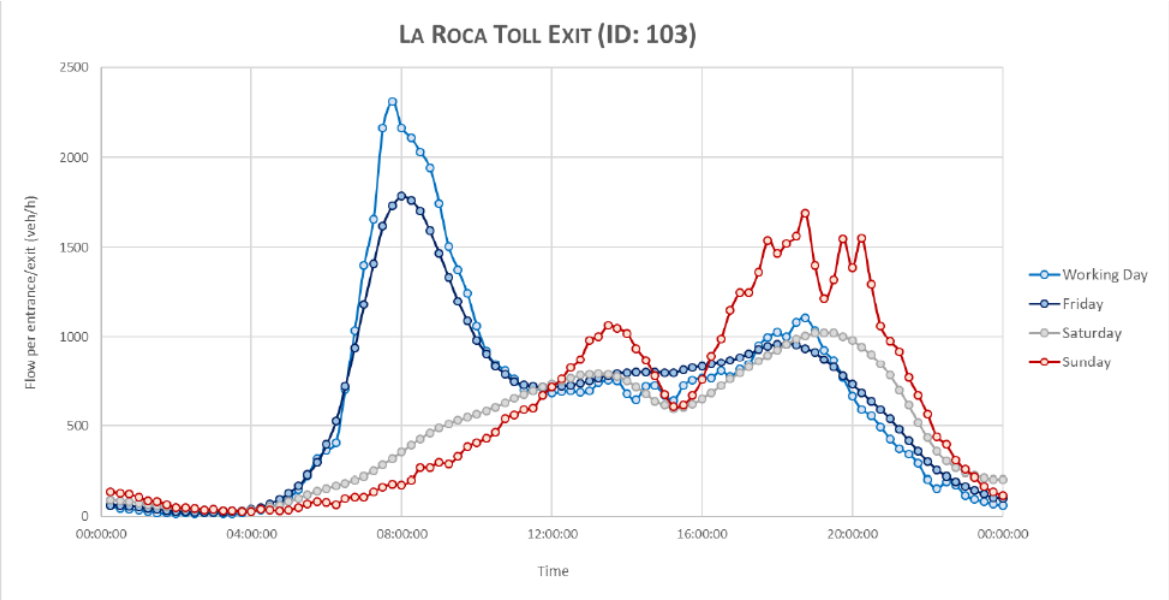


Figure 4.3: La Roca toll exit demand for the different traffic demands. Source: Aimsun [2019]

4.2 Scenarios, Replicas and Data Aggregation

This Section will explain which data has been used, how the data is prepared and the different traffic scenarios that take part in the performance of all the experiments in this thesis.

4.2.1 Scenarios

A simulation scenario is the combination of the traffic network and the traffic demand, which are explained at Section 4.1. The traffic network and the traffic demand used in this thesis are explained in Section 4.1, but in order to perform better experiments, another scenario has been created.

That scenario will have the same network but another demand. The demand will be the **WorkDay** demand multiplied by two in order to create a congested scenario. So the demand of this new scenario will be with 346.392 vehicles, which are performed with the following type of vehicles shown at Table 4.2.

Number of vehicles	**Type of vehicle**
Trucks	7.370
Trucks with Teletac	63.575
Cars	164.746
Cars with Teletac	110.700

Table 4.2: Vehicles for Congested WorkDay

4.2.2 Replicas

As explained in Section 4.1, the data is generated using a traffic simulation tool, Aimsun.

For every simulation scenario, we can create experiments that can have one or more replicas. Every replica has a seed, that determines the randomness within the simulation, such as the vehicle generations, the behaviour of the vehicles, the time that a vehicle appears and arrives, the speed of the vehicles, etc. Although this randomness is small, it is enough to see visible differences between the outputs of the two simulations.

Instead of using the same simulation for every day of data, every day is a different replica in order to ensure some difference between the days as happens in a real scenario.

4.2.3 Data Preparation

The data used in this masters thesis will be generated using Aimsun and considering two different scenarios as previously explained at subsection 4.2.1. Both scenarios are treated in the same way using the same exact code.

Aimsun simulator has an API which the user can interact. Using this API data can be retrieved, the behaviour of a vehicle modified, simulate an accident, etc. In this case, PVD has been extracted. As explained in Section 4.1, Aimsun is a microscopic traffic simulator, which means that simulates the behaviour of every vehicle in the scenario, so the data of every vehicle has been extracted at every simulation step which is by default 0.8s. The attributes of the data retrieved from Aimsun are:

- Vehicle ID: Identification of the vehicle inside Aimsun.
- Time: Time of the day inside the simulation in seconds. From 0s to 86400s(1 day).
- Speed: Speed of the vehicle in m/s.
- Section: Link where the vehicle is.
- Lane: Lane of the freeway where the vehicle is.
- Date: Date of the simulation inside Aimsun.

The simulations to generate the data take a lot of time. Each replica of the standard scenario takes 30 minutes and each replica of the congested scenario one and a half hour. Each replica outputs a data set with a size of 550MB for the standard scenario and a dataset of 1.7GB for the congested scenario.

The specifications of the C-Roads Spain project stand that every five minutes all the data of every vehicle will send together with an aggregate of those five minutes. This aggregation of the data is done by link, so every 5 minutes a summary of all the links in the model is sent. It makes more sense to use the aggregate data rather than the data of every vehicle because what is interesting is the state of a link rather than the state of a vehicle. So with the aggregation of the data, all the information about every link every five minutes is available.

The data extracted from Aimsun has been aggregated executing a Python script using Jupyter Notebook[5]. The Pandas[6] library has been a key part in the development of this aggregation algorithm as Pandas gives the developed valuable tools to manipulate data.

This data aggregation process computes the mean speed of every link, every aggregation time (1 minute, 5 minutes or 10 minutes) as the aggregation time is one factor of the proposed Design of Experiments as explained in Section 4.4.

Some replicas produce missing values when aggregated. This is caused because no connected cars went through that section s during the time interval of the aggregation t. This problem occurs mainly on sections that are very small, smaller than 50 meters. This problem is solved by setting the speed value of section s at time t to the mean of first past and first future recorded value.

[5]https://jupyter.org
[6]https://pandas.pydata.org

4.3 Goodness of Fit Measures

The performance of the algorithms will be measured using error measures. As the algorithms developed in this thesis forecast future values, the accuracy should be measured by summarizing the forecast errors. In this thesis, the main error measure used will be the Normalized Root Mean Squared Error (NRMSE).

The RMSE is probably one of the most common error measures used in regression problems. It is an error measure that penalizes big errors rather than the small ones. The units of the RMSE are the same units of the problem, seconds in this case as we are predicting travel time. So, the RMSE is used to measure the difference between the predicted values and the real values, which in this case, will be the values obtained by simulation using Aimsun.

$$RMSE_i = \sqrt{(X_{Aimsun,i} - X_{Forecasted,i})^2} \tag{4.1}$$

where:

i is the link.

$X_{Aimsun,i}$ is the value obtained through the Aimsun simulation

$X_{Forecasted,i}$ is the value forecasted by one of the models.

But, the problem of RMSE is that is difficult to interpret. NRMSE give the results in %, it is easier to interpret that one prediction has a 10% of error rather than a prediction has 30s of error. That is the reason behind using the NRMSE rather than the RMSE. Equation 4.2 show the NRMSE formula. The NRMSE will be computed for each section and after, an averaged NRMSE will be computed.

$$NRMSE_i = \frac{RMSE}{\overline{X_{Aimsun,i,t}}} \tag{4.2}$$

where:

RMSE is calculated within Equation 4.1

$\overline{X_{Aimsun,i,t}}$ is the mean travel time in the link i at the time of prediction t.

4.4 Design of Experiments

This section presents all the experiments performed in this master thesis with their results. The goal of these experiments is to test and validate how the algorithms that have been developed in this thesis work in different situations. In addition, the design of experiments performed in this thesis is described. Table 4.3 shows the different factors together with the levels for each factor. The proposed algorithms (ARIMA, RZ and

NN) explained at Chapter 3 have been tested in both scenarios (Normal and Congested). The rest of the factors are tested using reference values and then changing the level of the factor that is being studied. The reference values are for PVD 10%, aggregation 5 minutes, prediction horizon 30 minutes and data of 3 days.

The factors of the design of experiments are the following:

- Scenario: This factor constraints the scenario used for that experiment. The possible values are Normal or Congested.
- PVD Penetration: PVD penetration is a key part of this design of experiments as PVD penetration limits the amount of connected cars. The possible values are 10%, 25%, 50%, 75% or 100%.
- Aggregation: Data aggregation has been wildly discussed in Section 4.2. Aggregation factor tells the temporal interval which the data has been aggregated. The possible values for this factor
- Prediction Horizon: This factor constraints the horizon of the prediction. The possible values for this factor are 10 minutes, 30 minutes and 60 minutes.
- Data: The amount of data is key when creating data-driven solutions. This factor have three levels which are 1 day of data, 3 days of data and 5 days of data.

<table>
<tr><th>Algorithms</th><th>Scenario</th><th>PVD penetration</th><th>Aggregation</th><th>Prediction Horizon</th><th>Data</th></tr>
<tr><td rowspan="2">ARIMA</td><td rowspan="3">Normal</td><td>10%</td><td rowspan="2">1 minute</td><td rowspan="2">10 minutes</td><td rowspan="2">Low (1 Day)</td></tr>
<tr><td>25%</td></tr>
<tr><td rowspan="2">RZ</td><td>50%</td><td rowspan="2">5 minutes</td><td rowspan="2">30 minutes</td><td rowspan="2">Medium (3 Days)</td></tr>
<tr><td rowspan="2">Congested</td><td>75%</td></tr>
<tr><td>NN</td><td>100%</td><td>10 minutes</td><td>60 minutes</td><td>High (5 Days)</td></tr>
</table>

Table 4.3: Design of Experiments

All the experiments performed in this master thesis follow an encoding that codifies the parameters of that experiment. All the levels for all the factors can be found in Table 4.3. For example, the name of the experiment *res_ARIMA_30_Mid_Nor_5min_10* has seven codes which are the following:

- res: It is the abbreviation of "result". All the experiments start with res.
- ARIMA: It is the name of the method used, ARIMA in this case.
- 30: This value encodes the prediction horizon of the experiment, in this case, 30 minutes.

- Mid: It encodes the amount of data of the experiment. In this case three days.
- Nor: It is the acronym of normal and makes reference to the scenario used.
- 5min: Makes reference to the aggregation of the data, in this case, the data is aggregated every 5 minutes.
- 10: It is the encoding for the PVD, in this case, 10% of PVD is used.

4.4.1 Parameter Calibration

Most of the models developed in this masters thesis must be configured in order to obtain the desired results. In this subsection, the parameter calibration for every algorithm developed is described.

4.4.1.1 ARIMA parameter calibration

As explained in Section 3.1, the ARIMA has three parameters which must be calibrated, p, d and q. The parameters of the ARIMA have been calibrated using a grid search. A grid search is a process of performing hyper parameter tuning in order to determine the optimal values for a given model. In this case, the possible values for the parameters of the ARIMA models computed in the experiments of this master thesis are the following:

- p: 1, 2, 3, 4 and 5
- d: 0, 1 and 2
- q: 1, 2, 3, 4 and 5

As the computation of one ARIMA with fixed parameters is very fast (less than 10s) performing a grid search could assure us better results and in the future recompute the ARIMA models using the optimal parameters obtained using the grid search, but this is out of the focus of this masters thesis.

4.4.1.2 RZ parameter calibration

The RZ algorithm explained at Section 3.2 has one parameter to calibrate, which is the σ, that is a certain variance that the user needs to specify. The authors of the paper use ten minutes. In the experiments, the σ used will be the same that the authors use, ten minutes.

Also, the parameter δ which is the time-lag it is the same than the prediction horizon explained at the beginning of this section.

The time of prediction will be the 8 a.m because it is the time with the highest variance as shown in Section 4.2.

4.4.1.3 Neural Networks parameter calibration

Neural Networks, as explained in Section 3.3, have a bunch of parameters to be tuned. In this case, the parameters that have been tuned are the number of layers, the number of neurons per layer and the batch size.

The batch size, which fixes the amount of data for every batch, has been fixed to 60 when the aggregation is one minute, 12 when the aggregation is 5 minutes and 6 when the aggregation is 10 minutes. This is done in order to ensure that every batch will contain an hour of data.

The number of layers and the number of neurons have been calibrated for the base case previously explained.

Table 4.4 show the different layers and neurons tested with the results for the base case. In base of this results, the number of layers chosen will be 2 and the number of neurons 256.

Layers	**Neurons**	**NRMSE**
1	16	0.0269578
1	32	0.0267769
1	64	0.0252745
1	128	0.0253894
1	256	0.0242192
2	16	0.0268095
2	32	0.0250335
2	64	0.0254206
2	128	0.0256262
2	256	0.0239649
3	16	0.0283915
3	32	0.0266017
3	64	0.0277777
3	128	0.0286664
4	256	0.0277513
4	16	0.0300041
4	32	0.0311459
4	64	0.0311600
4	128	0.0316272
4	256	0.0304384

Table 4.4: Table of the tested parameters for the NN

4.5 Results

In this Section, the results of the experiments introduced in Section 4.4 are presented. For every factor explained in Section 4.4 there is a description of the experiments done, a Table with the NRMSE and plots that explain in a visual way the results obtained.

4.5.1 Hardware Used

The computer used to execute all the experiments has a Windows 10 as O.S and the following hardware:

- CPU Intel(R) Core(TM) i7-6700K CPU @ 4.00GHz.
- SSD Samsung 850 EVO M.2 500GB.
- 4x16GB RAM.
- GPU NVIDIA GeForce GTX TITAN X.

4.5.2 PVD Penetration Rate

In the near future, all the new cars will be connected cars. Connected cars have the ability to communicate with the infrastructure by sending important information such as speed, position, emissions, acceleration, vehicle state and many more. Nowadays is very difficult to imagine a scenario with a penetration rate of 100%. So, in this subsection the effect of the penetration rate of PVD will be analyzed using the simulated data described in Section 4.2 in order to conclude which penetration rate is the minimum in order to perform accurate forecasting using the three algorithms explained at Chapter 3.

Table 4.5 summaries the proposed experiments and details of the corresponding levels of the design factors. The experiments combine the five different PVD (10%, 25%, 50%, 75% and 100%) with the three algorithms (ARIMA, RZ and NN) together with the two proposed scenarios (Normal and Congested).

The data used in this experiment is data that have been aggregated using the process described in Section 4.2.

Name	Scenario	Algorithm	PVD
res_ARIMA_30_Mid_Nor_5min_10	Normal	ARIMA	10%
res_ARIMA_30_Mid_Nor_5min_25			25%
res_ARIMA_30_Mid_Nor_5min_50			50%
res_ARIMA_30_Mid_Nor_5min_75			75%
res_ARIMA_30_Mid_Nor_5min_100			100%
res_RZ_30_Mid_Nor_5min_10		RZ	10%
res_RZ_30_Mid_Nor_5min_25			25%
res_RZ_30_Mid_Nor_5min_50			50%
res_RZ_30_Mid_Nor_5min_75			75%
res_RZ_30_Mid_Nor_5min_100			100%
res_NN_30_Mid_Nor_5min_10		NN	10%
res_NN_30_Mid_Nor_5min_25			25%
res_NN_30_Mid_Nor_5min_50			50%
res_NN_30_Mid_Nor_5min_75			75%
res_NN_30_Mid_Nor_5min_100			100%
res_ARIMA_30_Mid_Con_5min_10	Congested	ARIMA	10%
res_ARIMA_30_Mid_Con_5min_25			25%
res_ARIMA_30_Mid_Con_5min_50			50%
res_ARIMA_30_Mid_Con_5min_75			75%
res_ARIMA_30_Mid_Con_5min_100			100%
res_RZ_30_Mid_Con_5min_10		RZ	10%
res_RZ_30_Mid_Con_5min_25			25%
res_RZ_30_Mid_Con_5min_50			50%
res_RZ_30_Mid_Con_5min_75			75%
res_RZ_30_Mid_Con_5min_100			100%
res_NN_30_Mid_Con_5min_10		NN	10%
res_NN_30_Mid_Con_5min_25			25%
res_NN_30_Mid_Con_5min_50			50%
res_NN_30_Mid_Con_5min_75			75%
res_NN_30_Mid_Con_5min_100			100%

Table 4.5: PVD Experiment Design

Table 4.6 shows the obtained results for the PVD factor. Figure 4.4 and Figure 4.5 show the plots for the normal and the congested scenario. Those figures show a clear trend that higher penetration ratio leads to better results. It is also remarkable how in the normal scenario, ARIMA and NN perform almost equally, but, in the congested scenario, NN performs better if % of PVD is higher than 10%. RZ is not affected by the penetration rate of PVD which is unexpected behaviour.

Name	**NRMSE**
res_ARIMA_30_Mid_Nor_5min_10	0.0371885
res_ARIMA_30_Mid_Nor_5min_25	0.0279149
res_ARIMA_30_Mid_Nor_5min_50	0.0262986
res_ARIMA_30_Mid_Nor_5min_75	0.0240389
res_ARIMA_30_Mid_Nor_5min_100	0.0249633
res_RZ_30_Mid_Nor_5min_10	0.1565731
res_RZ_30_Mid_Nor_5min_25	0.1565690
res_RZ_30_Mid_Nor_5min_50	0.1565721
res_RZ_30_Mid_Nor_5min_75	0.1565703
res_RZ_30_Mid_Nor_5min_100	0.1566052
res_NN_30_Mid_Nor_5min_10	0.0239944
res_NN_30_Mid_Nor_5min_25	0.3374594
res_NN_30_Mid_Nor_5min_50	0.0230887
res_NN_30_Mid_Nor_5min_75	0.0238514
res_NN_30_Mid_Nor_5min_100	0.0491041
res_ARIMA_30_Mid_Con_5min_10	0.0538967
res_ARIMA_30_Mid_Con_5min_25	0.0569831
res_ARIMA_30_Mid_Con_5min_50	0.0543924
res_ARIMA_30_Mid_Con_5min_75	0.0525718
res_ARIMA_30_Mid_Con_5min_100	0.0529864
res_RZ_30_Mid_Con_5min_10	0.1944981
res_RZ_30_Mid_Con_5min_25	0.1944625
res_RZ_30_Mid_Con_5min_50	0.1944688
res_RZ_30_Mid_Con_5min_75	0.1945003
res_RZ_30_Mid_Con_5min_100	0.1944269
res_NN_30_Mid_Con_5min_10	0.0877027
res_NN_30_Mid_Con_5min_25	0.3374594
res_NN_30_Mid_Con_5min_50	0.0937115
res_NN_30_Mid_Con_5min_75	0.0238514
res_NN_30_Mid_Con_5min_100	0.0491041

Table 4.6: PVD Results

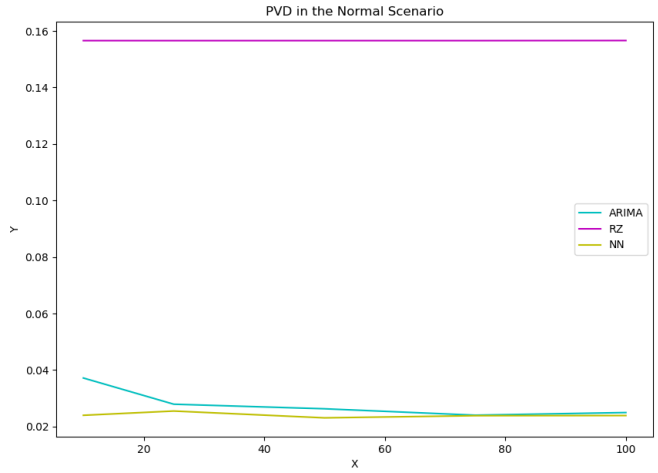


Figure 4.4: PVD in the normal scenario.

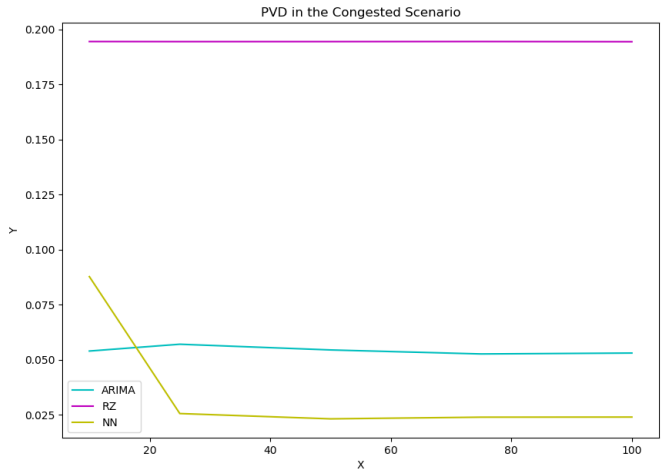


Figure 4.5: PVD in the congested scenario.

4.5.3 Amount of Data

One of the key factors to consider when forecasting is the amount of data available for that forecasting. In the near future, when a high percentage of vehicles are connected, it would be possible to have access to massive historical data. Nowadays though, this is not the scenario. The objective of this factor is to test which is the minimum amount of data needed for the correct forecasting of the algorithms developed explained at Chapter 3.

The proposed Design of Experiments for the Amount of Data factor is found in Table

4.7. The levels that have been chosen for this factor are Low(1 Day of data), Mid (3 Days of data) and Hi (5 days of data) that combine with the three algorithms (ARIMA, RZ and NN) together with the two proposed scenarios (Normal and Congested).

Name	**Scenario**	**Algorithm**	**Amount of data**
res_ARIMA_30_Low_Nor_5min_10	Normal	ARIMA	1 Day
res_ARIMA_30_Mid_Nor_5min_10			3 Days
res_ARIMA_30_Hi_Nor_5min_10			5 Days
res_RZ_30_Low_Nor_5min_10		RZ	1 Day
res_RZ_30_Mid_Nor_5min_10			3 Days
res_RZ_30_Hi_Nor_5min_10			5 Days
res_NN_30_Low_Nor_5min_10		NN	1 Day
res_NN_30_Mid_Nor_5min_10			3 Days
res_NN_30_Hi_Nor_5min_10			5 Days
res_ARIMA_30_Low_Con_5min_10	Congested	ARIMA	1 Day
res_ARIMA_30_Mid_Con_5min_10			3 Days
res_ARIMA_30_Hi_Con_5min_10			5 Days
res_RZ_30_Low_Con_5min_10		RZ	1 Day
res_RZ_30_Mid_Con_5min_10			3 Days
res_RZ_30_Hi_Con_5min_10			5 Days
res_NN_30_Low_Con_5min_10		NN	1 Day
res_NN_30_Mid_Con_5min_10			3 Days
res_NN_30_Hi_Con_5min_10			5 Days

Table 4.7: Amount of Data Experiment Design

Table 4.8 shows the obtained results for the amount of data factor. Figure 4.6 and Figure 4.7 show the plots for the normal and the congested scenario.

These results do not follow an intuitive trend, because it is intuitive to think that more data have to lead to better results. As the PVD penetration used for these experiments is 10%, these results prove that low penetration rates for 5 days or more with the filling of the missing values performed, explained at Section 4.2, tend to lead to worse results. It is also remarkable how the RZ algorithm in the congested scenario does not improve or worse at all, but for the Normal scenario seems to work better with data of the past 3 days instead of 5 days.

This issue could be solved by having more penetration of PVD.

Name	NRMSE
res_ARIMA_30_Low_Nor_5min_10	0.0392933
res_ARIMA_30_Mid_Nor_5min_10	0.0371885
res_ARIMA_30_Hi_Nor_5min_10	0.0920701
res_RZ_30_Low_Nor_5min_10	0.1818982
res_RZ_30_Mid_Nor_5min_10	0.1565731
res_RZ_30_Hi_Nor_5min_10	0.1821823
res_NN_30_Low_Nor_5min_10	0.0259011
res_NN_30_Mid_Nor_5min_10	0.0239944
res_NN_30_Hi_Nor_5min_10	0.0711858
res_ARIMA_30_Low_Con_5min_10	0.0378317
res_ARIMA_30_Mid_Con_5min_10	0.0538967
res_ARIMA_30_Hi_Con_5min_10	0.1021081
res_RZ_30_Low_Con_5min_10	0.1950751
res_RZ_30_Mid_Con_5min_10	0.1944981
res_RZ_30_Hi_Con_5min_10	0.1967008
res_NN_30_Low_Con_5min_10	0.0328240
res_NN_30_Mid_Con_5min_10	0.0877027
res_NN_30_Hi_Con_5min_10	0.0760951

Table 4.8: Amount of Data Results

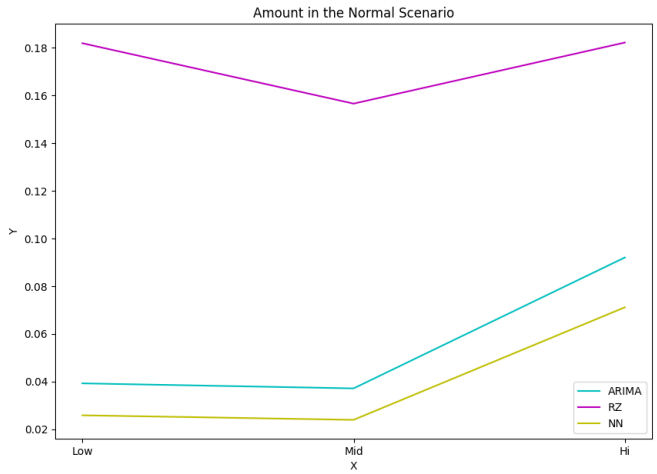


Figure 4.6: Amount of data in the normal scenario.

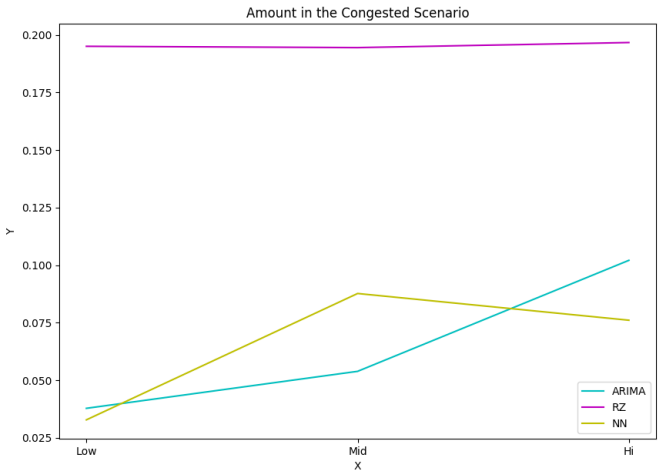


Figure 4.7: Amount of data in the congested scenario.

4.5.4 Aggregation

In Section 4.2 is explained that the data used in this thesis will be aggregated data that is obtained via the data extracted from Aimsun, explained at Section 4.1. One key factor to analyze is which aggregation is optimal for travel time forecasting in freeways using PVD.

In order to find the optimal aggregation, three levels of aggregation are purposed, which are one minute (1min), five minutes (5min) and ten minutes (10min). The Table 4.9 summarizes the experiments for this factor.

Name	Scenario	Algorithm	Aggregation
res_ARIMA_30_Mid_Nor_1min_10	Normal	ARIMA	1 minute
res_ARIMA_30_Mid_Nor_5min_10			5 minutes
res_ARIMA_30_Mid_Nor_10min_10			10 minutes
res_RZ_30_Mid_Nor_1min_10		RZ	1 minute
res_RZ_30_Mid_Nor_5min_10			5 minutes
res_RZ_30_Mid_Nor_10min_10			10 minutes
res_NN_30_Mid_Nor_1min_10		NN	1 minute
res_NN_30_Mid_Nor_5min_10			5 minutes
res_NN_30_Mid_Nor_10min_10			10 minutes
res_ARIMA_30_Mid_Con_1min_10	Congested	ARIMA	1 minute
res_ARIMA_30_Mid_Con_5min_10			5 minutes
res_ARIMA_30_Mid_Con_10min_10			10 minutes
res_RZ_30_Mid_Con_1min_10		RZ	1 minutes
res_RZ_30_Mid_Con_5min_10			5 minutes
res_RZ_30_Mid_Con_10min_10			10 minutes
res_NN_30_Mid_Con_1min_10		NN	1 minutes
res_NN_30_Mid_Con_5min_10			5 minutes
res_NN_30_Mid_Con_10min_10			10

Table 4.9: Aggregation Experiment Design

Table 4.10 shows the results for the aggregation factor and Figures 4.8 and 4.9 show the plots for the Normal and Congested scenarios. For the Normal scenario can be seen how the one-minute aggregation leads to very bad results for the ARIMA, more than 22% of error, but that goes down to the 2% when testing the 5 minutes aggregation. The NN follows that pattern but softer. Can also be seen how both ARIMA and NN get worse when increasing the aggregation to 10 minutes, that is because in the normal scenario (real demand) in 10 minutes something could change a significant enough to decrease the accuracy of the predictions. RZ algorithms perform equally in the three aggregation levels with an error of 15.5%.

Congested scenario is different for ARIMA and NN. The results for the one minute aggregation are also bad, but, the results for the 10 minute are better than the five minute. This happens because the demand of the Congested scenario is double the Normal, so the events in this scenario happen much slower than the Normal Scenario.

Name	NRMSE
res_ARIMA_30_Mid_Nor_1min_10	0.0755928
res_ARIMA_30_Mid_Nor_5min_10	0.0378122
res_ARIMA_30_Mid_Nor_10min_10	0.0840300
res_RZ_30_Mid_Nor_1min_10	0.1581572
res_RZ_30_Mid_Nor_5min_10	0.1565731
res_RZ_30_Mid_Nor_10min_10	0.1581572
res_NN_30_Mid_Nor_1min_10	0.2223325
res_NN_30_Mid_Nor_5min_10	0.0239944
res_NN_30_Mid_Nor_10min_10	0.0620530
res_ARIMA_30_Mid_Con_1min_10	0.0945414
res_ARIMA_30_Mid_Con_5min_10	0.0538753
res_ARIMA_30_Mid_Con_10min_10	0.0496989
res_RZ_30_Mid_Con_1min_10	0.1934264
res_RZ_30_Mid_Con_5min_10	0.1944981
res_RZ_30_Mid_Con_10min_10	0.1980316
res_NN_30_Mid_Con_1min_10	0.1156523
res_NN_30_Mid_Con_5min_10	0.0877027
res_NN_30_Mid_Con_10min_10	0.0402928

Table 4.10: Aggregation results

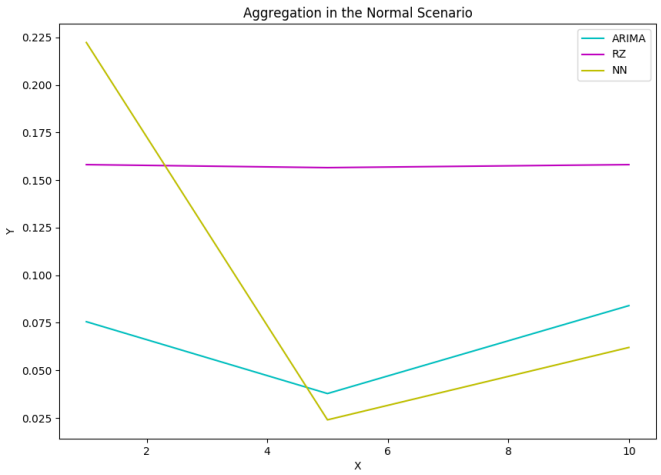


Figure 4.8: Aggregation in the normal scenario.

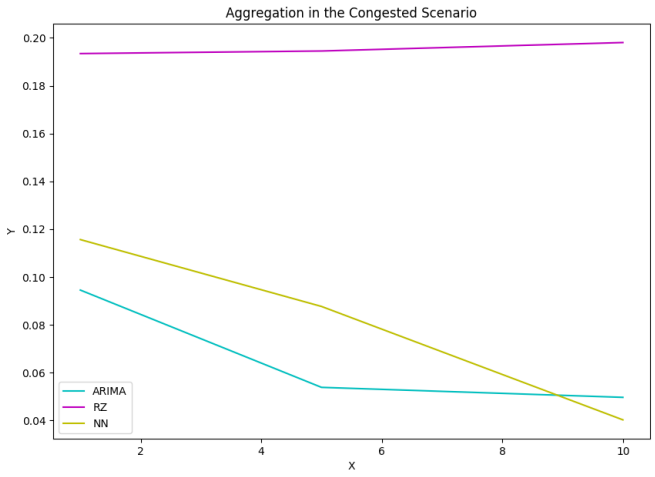


Figure 4.9: Aggregation in the congested scenario.

4.5.5 Prediction Horizon

When working with time series, a factor that is always crucial is to know how many ahead the prediction will be. The usual tendency of prediction horizon is very intuitive, larger the horizon, larger the error. The objective of this subsection is to explore in which prediction horizon the algorithms explained at Section 3 forecast with success.

Table 4.11 summarizes the experiment design for this factor. The prediction horizons that are considered are ten minutes(10), thirty minutes (30) and sixty minutes (60).

Name	Scenario	Algorithm	Prediction Horizon
res_ARIMA_10_Mid_Nor_1min_10	Normal	ARIMA	10 minutes
res_ARIMA_30_Mid_Nor_5min_10			30 minutes
res_ARIMA_60_Mid_Nor_10min_10			60 minutes
res_RZ_10_Mid_Nor_1min_10		RZ	10 minutes
res_RZ_30_Mid_Nor_5min_10			30 minutes
res_RZ_60_Mid_Nor_10min_10			60 minutes
res_NN_10_Mid_Nor_1min_10		NN	10 minutes
res_NN_30_Mid_Nor_5min_10			30 minutes
res_NN_60_Mid_Nor_10min_10			60 minutes
res_ARIMA_10_Mid_Con_1min_10	Congested	ARIMA	10 minutes
res_ARIMA_30_Mid_Con_5min_10			30 minutes
res_ARIMA_60_Mid_Con_10min_10			60 minutes
res_RZ_10_Mid_Con_1min_10		RZ	10 minutes
res_RZ_30_Mid_Con_5min_10			30 minutes
res_RZ_60_Mid_Con_10min_10			60 minutes
res_NN_10_Mid_Con_1min_10		NN	10 minutes
res_NN_30_Mid_Con_5min_10			30 minutes
res_NN_60_Mid_Con_10min_10			60 minutes

Table 4.11: Prediction Horizon Experiment Design

The Table 4.12 shows the results for the prediction horizon factor, also Figure 4.10 shows the results for the prediction horizon factor for the Normal scenario and Figure 4.11 shows the results of the prediction horizon for the Congested scenario.

The pattern of the ARIMA and NN are really intuitive. More time beyond the prediction is made, less accuracy on the prediction will be. This pattern can be seen both in the Congested and the Normal scenario. Normal scenario shows how the prediction get worse as the prediction horizon is increased but not that much as in the Congested scenario because in the congested scenario in one hour the vehicles can go from freeflow (33.33 m/s) to full stop due a traffic jam. RZ results also follow the same pattern in the congested scenario but with higher error rate but for the normal scenario show a very strange behaviour in the 10 minutes horizon.

Name	NRMSE
res_ARIMA_10_Mid_Nor_1min_10	0.0376169
res_ARIMA_30_Mid_Nor_5min_10	0.0378122
res_ARIMA_60_Mid_Nor_10min_10	0.0412305
res_RZ_10_Mid_Nor_1min_10	0.1826206
res_RZ_30_Mid_Nor_5min_10	0.1565731
res_RZ_60_Mid_Nor_10min_10	0.2041527
res_NN_10_Mid_Nor_1min_10	0.0221840
res_NN_30_Mid_Nor_5min_10	0.0239944
res_NN_60_Mid_Nor_10min_10	0.0301210
res_ARIMA_10_Mid_Con_1min_10	0.0375846
res_ARIMA_30_Mid_Con_5min_10	0.0496989
res_ARIMA_60_Mid_Con_10min_10	0.0751474
res_RZ_10_Mid_Con_1min_10	0.1926878
res_RZ_30_Mid_Con_5min_10	0.1944981
res_RZ_60_Mid_Con_10min_10	0.2041527
res_NN_10_Mid_Con_1min_10	0.0327470
res_NN_30_Mid_Con_5min_10	0.0877027
res_NN_60_Mid_Con_10min_10	0.1602395

Table 4.12: Prediction Horizon Results

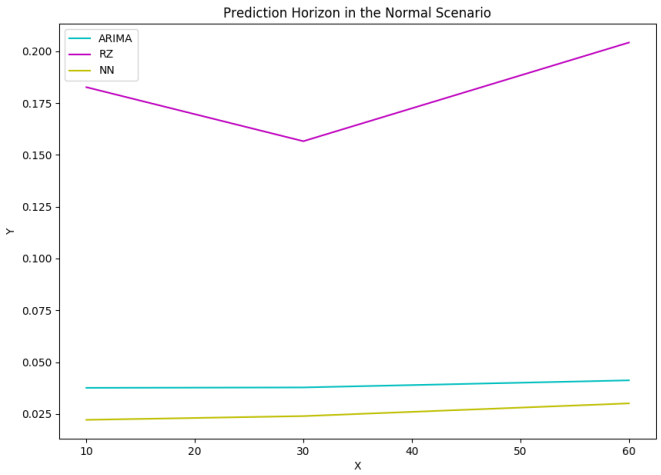


Figure 4.10: Prediction Horizon in the normal scenario.

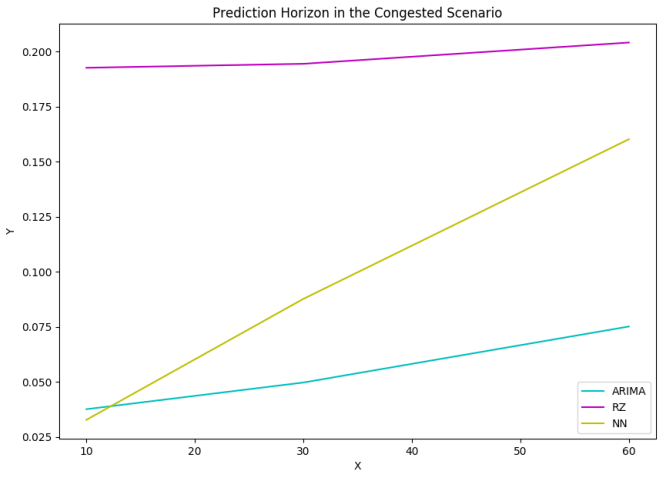


Figure 4.11: Prediction Horizon in the congested scenario.

Chapter 5

Final Comments

This Chapter shows the main achievements of this master thesis through paying special attention to the proposed objectives and research questions. The Chapter continues with a discussion of the contributions of this master thesis and further research.

5.1 Conclusions

This master thesis is focused on solving the problem of traffic forecasting on freeways using PVD. As it is shown in the state of the art at Chapter 2, this is a very extensive topic with a lot of research in it. This is because traffic forecasting is key in order to manage traffic properly.
The main objective of this master thesis is to perform traffic forecasting in freeways using PVD and to develop different machine learning algorithms in order to create a data-driven solution. In this master thesis, the proposed state of the art analyzed which methods and strategies are proposed in the literature to solve the problem. In particular, the literature shows how ARIMA, linear regression and Neural Networks are capable of efficiently solving it. Thus, the methods implemented in this master thesis are ARIMA, linear regression with varying parameters proposed by Rice and van Zwet [2004] (RZ) and a neural network.
We can conclude that the ARIMA and the NN forecast better than the RZ algorithms in both scenarios. Also, the penetration rate of PVD and the prediction horizon follow the expected pattern. However, the aggregation factor follows a different pattern depending on the scenario and the amount of data factor does not follow the expected pattern. More details about the obtained results for each of these methods can be found at Section 4.5.

On the following, we will answer the proposed research questions in Section 1.3:

Which of the machine learning solutions proposed is better in terms of prediction and efficiency? There is a method that is always the best?

We can conclude that NN and ARIMA have very similar results with respect to

NRMSE measure, however, the obtained results show that the method proposed by Rice and van Zwet [2004] does not give the expected results.

Does the Neural Networks (NN) perform better than the simpler models like ARIMA or Linear Regression?

As answered in the previous research question, NN outperforms the RZ algorithm but do not outperform ARIMA for all the performed experiments, in fact, they have very similar results.

How does the penetration of the PVD affects the traffic forecasting in freeways? Which is the minimal penetration needed in order to secure the predictions of a data-driven solution?

The accuracy of ARIMA and NN increases when the penetration rate of PVD is increased too. Thus, PVD penetration probed to be a key feature in travel time forecasting using PVD. Also, it can be stated that the minimum PVD penetration for NN is 25%. ARIMA with a 10% of PVD gave better results than NN however, in higher PVD ratios NN outperforms ARIMA.

How the prediction horizon affects the overall forecasting?

As expected, the performed experiments show that the prediction horizon factor strongly affects the accuracy of the forecasting methods developed in this master thesis. Mainly, the results prove that larger horizons lead to worse predictions. Nevertheless, depending on the scenario the observed effect is more or less significant.

What is the amount of data (in days) needed to perform accurate forecasting?

The amount of data factor does not have the expected behaviour due to the design of experiments performed because the reference value of PVD on these experiments is 10%. I would be interesting to perform more experiments using different amounts of data and also changing the percentage of PVD. However, the results for the normal scenario show better results when using data of the past three days.

Does the aggregation of the data affects the forecasting in a significant way?

The results show how aggregation has proven to be a key factor in this master thesis. The results of the aggregation factor state that the best aggregation will depend on traffic demand. Congested scenarios change slower than less congested scenarios, that could be the reason why the congested scenario works better with larger aggregation and the normal scenario work better with smaller aggregations. Although, aggregations of one minute gave bad results in both scenarios.

5.2 Contributions

In addition to the goals archived and the answered research questions in this master thesis, the contributions of this master thesis are also presented in this Section.

The first contribution is a state of the art about traffic forecasting in freeways that can be found at Chapter 2. This state of the art makes special emphasis on the traffic forecasting using PVD.

Another contribution is the adaptation of the Rice and van Zwet [2004] (RZ), changing the approach of the research of the authors from a loop detector perspective to PVD.

In addition, the development of ARIMA, RZ and NN are also a contribution of this master thesis. The data aggregation algorithms that make possible the input for those algorithms can also be used to create the input for other methods such as LSTM or RNN.

Also, the development of a simulation-based framework to extract data from Aimsun and also to test the different forecasting algorithms with different scenarios and situations. This contribution can also be used for further research on this topic adding more algorithms, networks or scenarios.

Finally, the traffic forecasting algorithms developed in this master thesis are a contribution to C-Roads, mobility European Project.

5.3 Further Research

Even the research questions have been answered and the proposed objectives have been archived, this master thesis opens new research lines to extend the work done. This Section exposes further research that could continue the work done in this master thesis. First of all, performing more experiments for the proposed factors with a factorial design of experiments. Also, increasing the levels of the proposed factors. For example, testing wider aggregations or longer prediction horizons.
Secondly, generate and use more training data than the proposed in the design of experiments at Section 4.4. It would be interesting to test these algorithms, especially the Neural Network with more data like one month of data. Also, the inclusion of all demand profiles described at Section 4.1 in the training data, not just the **WorkDay** demand profile. The inclusion of these methods could lead to the implementation of new methods such an RNN or LSTM. As shown in Chapter 2, this research field is very active and new solutions are proposed constantly. And in the near future, with more computational power, would be possible to use even more training data with algorithms that are more computationally expensive.

In addition, the usage of real connected car data is a very interesting research line and a technological challenge. Testing the algorithms developed in this master thesis in a 100% real scenario would be the next step for the C-Roads project.

Finally, the implementation of the algorithms in a cloud platform such as Google

Cloud or AWS could be a very interesting proposal with the needed adaptation of these procedures done in this master thesis into the real C-Roads scenario.

Bibliography

Travel time estimation of a path using sparse trajectories. *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '14*, (5):25–34, 2014. doi: 10.1145/2623330.2623656. URL http://dl.acm.org/citation.cfm?doid=2623330.2623656.

Aimsun. *C-Roads Aimsun Network Simulation Models AP7*. Aimsun SLU, 2019.

Colah. Understanding lstm networks. URL https://colah.github.io/posts/2015-08-Understanding-LSTMs/.

F.M. Sanders C.P.IJ. van Hinsbergen, J.W.C. van Lint. Deliverable DIIF-1a. (February), 2007.

engMRK. Gradient descent with momentum. URL https://engmrk.com/gradient-descent-with-momentum/.

Erik Jenelius and Haris N. Koutsopoulos. Travel time estimation for urban road networks using low frequency probe vehicle data. *Transportation Research Part B: Methodological*, 53:64–81, 2013. ISSN 01912615. doi: 10.1016/j.trb.2013.03.008. URL http://dx.doi.org/10.1016/j.trb.2013.03.008.

Erik Jenelius and Haris N. Koutsopoulos. Urban network travel time prediction based on a probabilistic principal component analysis model of probe data. 19(2):1–23, 2016. doi: 10.13140.

Ibai Lana, Javier Del Ser, Manuel Velez, and Eleni I. Vlahogianni. Road Traffic Forecasting: Recent Advances and New Challenges. *IEEE Intelligent Transportation Systems Magazine*, 10(2):93–109, 2018. ISSN 19411197. doi: 10.1109/MITS.2018.2806634.

Xiaolei Ma, Zhimin Tao, Yinhai Wang, Haiyang Yu, and Yunpeng Wang. Long short-term memory neural network for traffic speed prediction using remote microwave sensor data. *Transportation Research Part C: Emerging Technologies*, 54:187–197, 2015. ISSN 0968090X. doi: 10.1016/j.trc.2015.03.014. URL http://dx.doi.org/10.1016/j.trc.2015.03.014.

Aditya Narayanan, Nikola Mitrovic, Muhammad Tayyab Asif, Justin Dauwels, and Patrick Jaillet. Travel Time Estimation Using Speed Predictions. *2015 IEEE 18th International Conference on Intelligent Transportation Systems*, pages 2256–2261, 2015. doi: 10.1109/ITSC.2015.364. URL `http://ieeexplore.ieee.org/document/7313456/`.

Daiheng Ni and Haizhong Wang. Trajectory reconstruction for travel time estimation. *Journal of Intelligent Transportation Systems: Technology, Planning, and Operations*, 12(3):113–125, 2008. ISSN 15472450. doi: 10.1080/15472450802262307.

J. Rice and E. van Zwet. A simple and effective method for predicting travel times on freeways. *IEEE Transactions on Intelligent Transportation Systems*, 5(3):200–207, Sep. 2004. ISSN 1524-9050. doi: 10.1109/TITS.2004.833765.

John Rice and Erik Van Zwet. A simple and effective method for predicting travel times on freeways. *IEEE Transactions on Intelligent Transportation Systems*, 5(3):200–207, 2004. ISSN 15249050. doi: 10.1109/TITS.2004.833765.

Bin Sun, Wei Cheng, Prashant Goswami, and Guohua Bai. An Overview of Parameter and Data Strategies for k-Nearest Neighbours Based Short-Term Traffic Prediction. pages 68–74, 2018. doi: 10.1145/3157737.3157749.

T.Hastie and R. Tibshirani. Varying coefficient models. *Journal of the Royal Statistical Society*, 55:757–796, 1994.

TowardsDataScience. The mostly complete chart of neural networks, explained. URL `https://towardsdatascience.com/the-mostly-complete-chart-of-neural-networks-explained-3fb6f2367464`.

Martin Treiber and Arne Kesting. Traffic flow dynamics: Data, models and simulation. *Traffic Flow Dynamics: Data, Models and Simulation*, 12:1–503, 2013. ISSN 15529924. doi: 10.1007/978-3-642-32460-4.

Wikipedia. Linear regression. URL `https://en.wikipedia.org/wiki/Linear_regression`.

Haofan Yang, Tharam Dillon, Yi-ping Chen, Haofan Yang, Tharam Dillon, Yi-ping Chen Evaluation, Computational Approaches, Haofan Yang, Tharam S Dillon, and Yi-ping Phoebe Chen. Evaluation of Recent Computational Approaches in Short-Term Traffic Forecasting To cite this version : HAL Id : hal-01383959 Evaluation of Recent Computational Approaches in Short-term Traffic Forecasting. 2016.

Hongsuk Yi, Jung Heejin, and Sanghoon Bae. Deep Neural Networks for traffic flow prediction. *2017 IEEE International Conference on Big Data and Smart Computing, BigComp 2017*, pages 328–331, 2017. ISSN 2375-933X. doi: 10.1109/BIGCOMP.2017.7881687.